Atom Nieuws

JAARGANG : 8

NUMMER : 2

DISKNRS. : 3-5 + MS-DOS NR.1 1989



Bestuur:

Voorzitter:	Secretaris:	Penningmeester:
P. van Kuik	J.Hartog	T.Rutten
Zuideinde 54-a	Keyenbergseweg 60	Berkenlaan 24
1843 JP Groot-Schermer	6871 WK Renkum	3737 RN Groenekan
Tel.02997-1902	Tel.08373-13757	Tel.03461-3495

Clubwinkel:	Redactie Atom Nieuws:	Atomtel:
P.Grevelt	E.Sanders	
Emmastraat 22	Rosslag 13	databank van de
1782 PD Den Helder	6049 BE Herten	ATOM club
Tel. 02230-23453	Tel. 04750-30401	Tel. 020-167383

Contributie 1989 : fl. 25.00 : Atom Computer Club : Giro 5244293

Redactie Atom Nieuws		SPS-Printservice	Ledenadministratie
E.Sanders	04750-30401	E.Sanders	S.van Leeuwen
B.Tossaint	043-431675	Rosslag 13	Kompasstraat 52
N.Schreurs	04490-73679	6049 BE Herten	1973 PX IJmuiden
R.Leurs	04490-36454	04750-30401	Tel. 02550-22435
W.Truyen	09-3211564792	Zie blz. 44	

Uiterste datum inlevering KOPIJ: nr. 8-3 15 Augustus 1989

De clubwinkel:

80-koloms videokaart	fl. 40.00
Geheugenkaart 16 kByte . exclusief onderdelen	fl. 1.00
(NEE, die fl. 1,-- is geen TYPfout!!!)	
Schakelkaart: deze is uitverkocht	
Minischakelkaart: ook deze kaart is uitverkocht	
Combikaart: Zie SPS Print-Service	
Z-80 kaart voor CP/M ,exclusief onderdelen	fl.100.00
ACORN NIEUWS 1982 , 97 pagina's samenvatting	fl. 2.50
ATOM NIEUWS jaargang 1983, ruim 450 pagina's	fl. 2.50
ATOM NIEUWS jaargang 1984, ruim 650 pagina's	fl. 2.50
ATOM NIEUWS jaargang 1985, ruim 650 pagina's	fl. 2.50
ATOM NIEUWS Jaargang 1986, ruim 500 pagina's	fl. 2.50
ATOM NIEUWS Jaargang 1987, ruim 300 pagina's	fl. 2.50
ATOM-WARE 1 : ATOM Basic interpreter , 98 pagina's	fl. 1.00
ATOM-WARE 2 : ATOM Disk Operating System , 68 pagina's	fl. 1.00
ATOM-WARE 3 : ATOM Monitor Operating System , 80 pagina's	fl. 1.00

LEVERING geschiedt via uw regionale penningmeester of rechtstreeks via de federatieve penningmeester . Bij rechtstreekse bestelling stort u het bedrag van het gewenste artikel , vermeerderd met fl.4.00 portokosten , op de giro van de federatie onder vermelding van de naam van het artikel en uw lidmaatschapsnummer.

AN-2 1989

ATOM NIEUWS is een uitgave van de federatie Atom computerclub Ned/Belgie en verschijnt 4 keer per jaar.

Het voorplaatje is weer afkomstig van Patrik [illegible] uit Roermond

VAN DE REDACTIE.

Deze maal allereerst enig onplezierig nieuws ; weliswaar verwacht,begrijpelijk, aangekondigd en voorzien,maar, daarom niet minder vervelend voor ons.
Jan Bronzwaer stopt met het zeer stevig,decoratief,maar ook indringend vasthouden van de redactie-dirigeerstok,-potlood, en - "computer ", zozeer zelfs ,dat hij inmiddels geen ATOM meer in zijn bezit heeft.
Wie de vele artikelen,ontwikkelingen,de kwaliteit van ACORN-NIEUWS -voorzover dat althans door de redactie wordt gemaakt-,in de laatste 6 afleveringen heeft gevolgd ; danwel schriftelijk of telefonisch vragen en/of verzoeken van hem kreeg,heeft gemerkt "dat hij er met hart en ziel bij was".
Met een elan en een grondigheid ,die ons,andere leden van de redactie, toch ook wel eens ontmoedigde.
Daarvoor,namens de redactie,maar ,ik dank ook namens alle leden ATOM-isten ,Jan ,onze hartelijke dank.

Het plezierige nieuws is ; het feit dat dit exemplaar van AN jullie nog voor de zomer-vacantie bereikt.
Het zag er namelijk enige tijd naar uit ,dat dit niet wilde lukken wegens gebrek aan copy. !!

Communicatie;koppeling aan andere computers ;vernieuwing van de ATOM, gebruik makend van de nu beschikbare,betaalbare hardware ; meer specialistische toepassingen van de ATOM ;zijn onderwerpen die al vaker speelden,maar ook in de toekomende tijd voldoende ruimte voor creativiteit bieden.

Schroom niet, om plannen, maar vooral ook problemen bij de ontwikkeling en realisatie van hard- en software via AN aan andere gebruikers mee te delen ,zodat ook het 3e nummer van dit jaar gevuld kan worden.

DE REDACTIE.

1-4-1989 PAGE: 1

ATOM-NIEUWS REGIO-SOFTWARE '89

dnr	program	gebruiken#	utility	soort	artikel	info
03	AD/BD.2	9A00-9FAF	--	DOSCONV	AN.8-2	CONVERSIEPROGRAMMA
03	ASS-LOG	7000-7FDD	--	COMMUN.	AN.8-2	VIDITEL:GEASS.INLOGCODE
03	ATOMAMP	7800-7BB4	--	ELECTR.	AN.8-2	LINEAN2:HULPFILE
03	BANDDRL	9800-9081	--	ELECTR.	AN.8-2	LINEAN:HULPFILE
03	DEMO84	3A00-3B96	P-CHARME	GRAFICS	AN.8-2	GDUMP84:DEMO-AANMAAK
03	DIR.AT#	2900-3440	P-CHARME	UTILITY	AN.8-2	TABEL MAKER P-CHARME
03	DISYC	2900-29FF	--	UTILITY	AN.8-2	GEFORMAT.LISTINGS
03	DUMP	5000-5FFF	--	UTILITY	AN.8-2	RS232 ATOMTEL UTILITY
03	ED80.v4	A000-AFFF	--	OSSOFT	AN.8-2	ED80-TEKSTVERW.
03	ED80src	7900-7ED5	--	OSSOFT	AN.8-2	SOURCE ED80
03	ERASbas	2900-2E74	P-CHARME	OSSOFT	AN.8-2	DISK-UTILITY
03	FLIST	2900-3307	--	UTILITY	AN.8-2	GEFORM.LISTINGS
03	GDUMP84	2900-3829	SALFAA	GRAFICS	AN.8-2	GKT-DUMPPROGRAMMA
03	INIT-LO	3A6C	P-CHARME	COMMUN.	AN.8-2	VIDITEL:AANMAK.INLOGCODE
03	LINEAN2	2900-35D4	GAGS	ELECTR.	AN.8-2	LINEAN
03	LISTER	2900-2969	--	UTILITY	AN.8-2	RS232 ATOMTEL
03	LOGIN	2900-37EF	SALFAA	COMMUN.	AN.8-2	VIDITEL:HULPPROGRAMMA
03	MENUbas	2900-2CD4	P-CHARME	OSSOFT	AN.8-2	OPSTART MENU
03	MODEM	2900-2BFF	P-CHARME	COMMUN.	AN.8-2	VIDITEL:AUT.INLOGGEN
03	OPEL04	0000-9800	--	PLAATJE	AN.8-2	GDUMP84:DEMO-FILE
03	PC-ATOM	2900-2E1F	--	COMMUN.	AN.8-2	PC AAN ATOM
03	RTTY	6000-7700	--	COMMUN.	AN.8-2	RTTY PROGRAMMA
03	SQUENIS	2900-3087	GAGS	SPEL	AN.8-2	SNEL EN MOOI
03	SQUEpic	8000-9800	--	SPEL	AN.8-2	SQUEENIS:SPELPLAATJE
03	SQUdemo	0000-7900	--	SPEL	AN.8-2	SQUEENIS:PLAATJE
03	T-RESON	9800-9BAF	--	ELECTR.	AN.8-2	LINEAN2:HULPFILE
03	WILHELM	2900-2C39	JOSBOX	MUZIEK	AN.8-2	NATIONALE HYMNE OP atom
04	A.B.C.	8200-95FF	--	UTILITY	AN.8-2	KRIMP-PROGRAMMA
04	BEWERK	1000-12DA		COMMUN	AN.8-2	BASLOAD:HULP
04	DCATado	8200-9400	--	DATBAS	AN.8-2	DISKCAT ATOMDOS
04	DCATgdo	8200-9400	--	DATBAS	AN.8-2	DISKCAT GDOS
04	DEMO#	2900-2B6C	--	CURSUS	AN.8-2	BASIC CURSUS
04	FUNEEM	2900-2F6B	P-CHARME	SPEECH	AN.8-2	KLANKGAMMA
04	INCODE1	0600-0A30	--	DATBAS	AN.8-2	INFO80:CODE
04	INFO801a	8000-9FFF	--	DATBAS	AN.8-2	INFOMASTER
04	INFO801m	0000-7FFF	--	DATBAS	AN.8-2	INFOMASTER:LADER
04	ITALfnt	5400-61ED	-	GRAFICS	AN.8-2	WRITE-III
04	KAARS	2900-3C09	P-CHARME	GRAFICS	AN.8-2	MOOI!
04	MENUasI	0600-0641	--	DATBAS	AN.8-2	INFOMASTER:MENU
04	NORMfnt	5400-6270	--	GRAFICS	AN.8-2	WRITE-III
04	PADDING	2900-3A90	--	ELECTR.	AN.8-2	BEREKENINGEN
04	SHADfnt	5400-61DE	--	GRAFICS	AN.8-2	WRITE-III
04	SPRINGO	2900-3005		SPEECH	AN.8-2	SPREKENDE BINGO
04	SPEECH	2900-3E11	--	SPEECH	AN.8-2	SPREKENDE ATOM
04	SPRDOOL	2900-3A24	P-CHARME	SPEL	AN.8-2	SPREKENDE DOOLHOF
04	STETfnt	5400-6264	GAGS	GRAFICS	AN.8-2	WRITE-III:FONTS
04	TB-TEST	2900-29FB	--	SPEECH	AN.8-2	TIMEBOX:DEMO
04	TIMEBOX	2900-4105	P-CHARM	SPEECH	AN.8-2	SPREKENDE KLOK
04	VDU-CGS	2900-3E15	P-CHARME	VDU	AN.8-2	KARAKTERSET

1-4-1989 PAGE: 2

ATOM-NIEUWS REGIO-SOFTWARE '89

dnr	program	geheugen#	utility	soort	artikel	info
04	VERG$	8200-8334	--	CURSUS	AN.8-2	BASIC CURSUS:VERGELIJK
04	WRITE-3	2900-4700	--	GRAFICS	AN.8-2	WRITE-III
04	WRITmch	5200-53BA	--	GRAFICS	AN.8-2	WRITE-III:MACH.TAAL
04	WRITpic	8000-9800	--	GRAFICS	AN.8-2	WRITE-III
04	1	2900-297F	--	DATBAS	AN.8-2	DISKCAT:LOADER
04	startad	2900-2A0D	--	DATBAS	AN.8-2	DISKCAT
04	startgd	2900-2A8Y	--	DATBAS	AN.8-2	DISKCAT
05	BASILex	2900-2077	--	COMMUN.	AN.8-2	UITLEG
05	BOOTREM	2900-2FCB	--	UTILITY	AN.8-2	BOOTEN MET DISKDIR.
05	CALCsrc	2900-7087	--	DATBAS	AN.8-2	SOURCE CALC-ROM
05	DB3	8FD0-B000	--	DATBAS	AN.8-2	DBASE
05	DB3-1	2900-5E16	--	DISASS	AN.8-2	DBASE3-SOURCE 1
05	DB3-2	2900-67D1	--	DISASS	AN.8-2	DBASE-SOURCE 2
05	DB3-ST	2900-2BF1	--	DATBAS	AN.8-2	VERBETERING DB3
05	RADILAD	1000 15FD	--	COMMUN.	AN.8-2	BASICODE LADER

MS-DOS schijf behorende bij AN-2 1989 bij de artikelen:
PC aan de Atom en RTTY TX/RX

ASS	BAT	593	26-03-89	1:00
BASLOAD	EXE	4784	26-03-89	1:00
BASLOAD	PAS	1980	26-03-89	1:00
CSC6502	EXE	21190	30-11-86	13:45
PC-ATOM	BAS	1407	26-03-89	1:00
RTTY	BAS	3224	26-03-89	2:00
RTTY	BIN	1793	26-03-89	2:00
RTTY	LST	53599	26-03-89	2:00
RTTY	MXT	8071	31-05-89	20:02
RTTY	SRC	21584	26-03-89	2:00
S1LOAD	EXE	10574	20-03-89	20:04

****
* SPEECH SYNTHESISER AAN DE ATOM *
****

Alles op een rijtje gezet door Roland Leurs.

Als U zo door de jaargangen van Atomnieuws bladert, dan ziet U dat er ieder jaar iets aan gedaan is om de Atom te laten spreken. Er zijn enkele IC's toegepast om dit te verwezenlijken. Het meest gebruikte IC is de SP-0256AL2 die voor ongeveer f 35,- te krijgen is. Met enkele komponenten erbij kunt U hier heel mooie resultaten mee behalen.

De verbinding tussen de Atom en het spraak ic kan op verschillende manieren gebeuren. Een 6522 VIA is in ieder geval noodzakelijk. De nodige aansluitingen zijn 6 datalijnen (A1 t/m A6), een strobe en een busy lijn. Deze lijnen kunnen we realiseren met de B-poort van de via. Dit heeft men in het verleden ook gedaan en er bestaat dus ook de nodige software hiervoor.

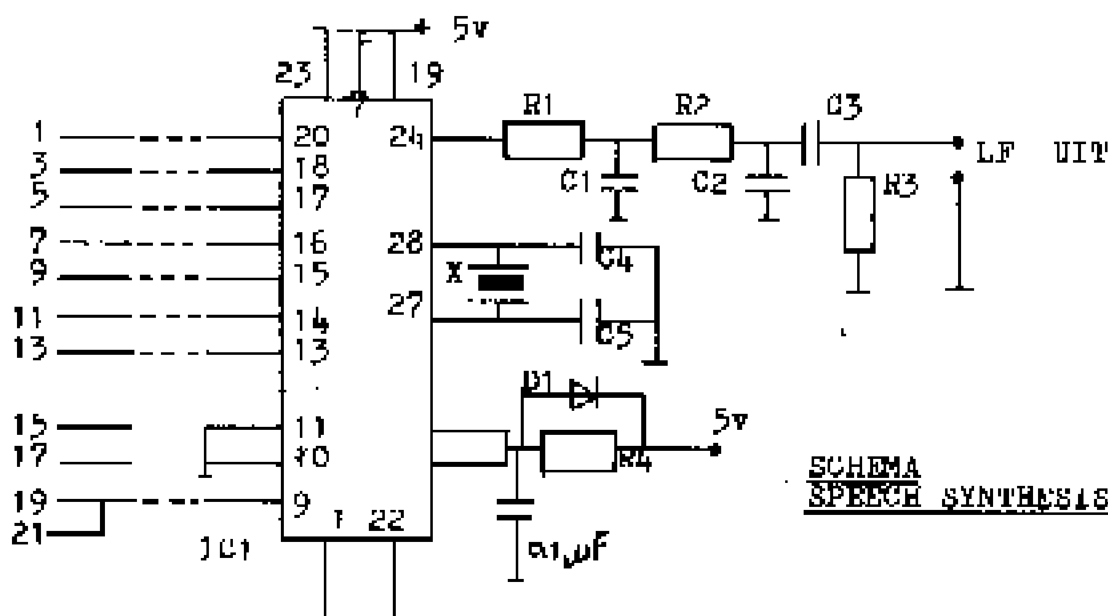
Echter in AN5.1 staat het idee om deze verbinding te maken met behulp van de printeraansluiting van de Atom. Dit heeft als voordeel dat de kabel meters lang mag zijn, de print eenvoudig blijft en de software is ook heel eenvoudig, bijvoorbeeld : P.$2$20$2$6$58$11$0$3 spreekt het woord 'ACORN' uit. Een ander voordeel van deze aansluiting is dat het geheel ook werkt op andere computers, bijv. een Electron.

Als U naar het schema kijkt ziet U dat er weinig onderdelen bijkomen. De geschatte bouwkosten zijn ongeveer f 50,-. Het kristal mag varieren van 3,12 tot 4 Mhz. Ik zelf gebruik een kristal van 3,2768 Mhz. Neem als het kan een waarde die zo dicht mogelijk bij 3,12 Mhz ligt.
Verder levert de bouw van het printje geen problemen op, zorg wel dat U de juiste rij van de printeraansluiting neemt anders werkt het geheel (uiteraard) niet. Zelf heb ik het geheel op een stuk gaatjesprint gemonteerd, de afmetingen kunnen zo klein gehouden worden dat de print in het doosje van een cassettebandje past!

Aan de uitgang van de schakeling wordt een versterkertje aangesloten. Dit kan van alles zijn. Mijn Atom behaalt goede resultaten met het bijgeleverde schema van een versterkertje. Als luidspreker gebruikt ik de mini-boxen voor mijn walkman.. Gaat prima !

Onderdelenlijst :

Speechprint:		Versterkertje:	
IC	: SP-0256 AL 2	IC2,3	: LM 386
R1,2	: 33 k	R5	: 29 k
R3	: 10 k	C6,7	: 220 uF
R4	: 100 k	C8	: 10 uF
C1,2	: 22 uF	C9	: 47 uF
C3	: 1 uF	P1	: 1 M instelpot.
C4,5	: 22 pF		
X	: 3,12 ... 4 Mhz		



SCHEMA
SPEECH SYNTHESISER

5v
0
LF IN
3
2
6
4
5
IC2
C6
C9
C8
6
3
8
2
4
5
IC3
P1
R5
C7
4 Ohm
1 W

SCHEMA VERSTERKERTJE

SOFTWARE

Er is ook al genoeg software voor het spraak io. Op de regioschijf staan de volgende programma's:

SPEECH SPBINGO - TIMEBOX - FONEEM - SPRDOOL

SPEECH is een programma van Ronald Boers, Theo den Exter en Bram Poot en staat beschreven in AN5.1 blz 44. Het programma maakt twee P Charme statements om allofonen naar de speech print te sturen.
Het statement SPEAK maakt 256 bytes vrij voor een speechbuffer waar de uit te spreken data tijdelijk opgeslagen worden. De syntax is SPEAK adres. Ook is dit mogelijk in een programma: DIM A(256);SPEAK A. Daarna kunt U rustig verder DIMmen.
Met SAY kunnen de klanken daarna verstuurd worden. De syntax van SAY is : SAY <foneem>,<foneem>.... De fonemen vindt U in de tabel op de volgende bladzijde. Voor meer mogelijkheden van SPEAK en SAY verwijs ik naar bovengenoemd artikel.

SPBINGO is een programma van J.P. Tap en staat ook in AN5.1 blz 70. Dit is een programma om een bingo-molentje te vervangen. Patrick Selder paste het programma aan zodat de getallen ook uitgesproken worden. De statements SPEAK en SAY worden hierbij gebruikt.

TIMEBOX is een programma dat ik zelf geschreven heb. Dit is een verzameling statements om een real time klok op interrupt basis te besturen. De tijd verschijnt rechtsboven in beeld of kan m.b.v. het statement TIME uitgesproken worden. Uitgebreide uitleg vindt U elders in dit blad.

FONEEM is een programma van Wim Schoenmakers. Dit programma vraagt om een reeks fonemen die later uitgesproken worden.

SPRDOOL is de doolhof uit het Atom Magic Book. Natuurlijk ook geschikt voor de speech synthesizer. Instrukties worden U toegesproken. Ook hier worden SPEAK en SAY gebruikt.
Dit programma geeft een leuk voorbeeld hoe de spraak in spelletjes kan worden verwerkt. Een nadeel is echter het engelse accent als de CP nederlands spreekt.

Nogmaals voor alle duidelijkheid: deze programma's werken alleen met de speechprint aan de printerpoort !

Lijst van fonemen :
(De hoofdletters zijn de betreffende klank)

00	00	PA1	PAUSE	10	20	32	AW	OUt	250
01	01	PA2	PAUSE	30	21	33	DD2	Do	250
02	02	PA3	PAUSE	50	22	34	GG3	wiG	120
03	03	PA4	PAUSE	100	23	35	VV	Vest	130
04	04	PA5	PAUSE	200	24	36	SH	Guest	80
05	05	OY	boY	290	25	37	SH	SHip	120
06	06	AY	skY	170	26	38	ZH	aZure	130
07	07	EH	End	50	27	39	RR2	bRain	80
08	08	KK3	Comb	80	28	40	FF	Food	110
09	09	PP	Pow	150	29	41	KK2	sKy	140
0A	10	JH	dodGe	100	2A	42	KK1	Can't	120
0B	11	NN1	thiN	170	2B	43	ZZ	Zoo	150
0C	12	IH	sIt	50	2C	44	NG	aNchor	200
0D	13	TT2	To	100	2D	45	LL	Lake	80
0E	14	RR1	Rural	130	2E	46	WW	Wool	140
0F	15	AX	sUcceed	50	2F	47	XR	repAIR	250
10	16	MM	Milk	180	30	48	WH	WHig	150
11	17	TT1	parT	80	31	49	YY1	Yes	90
12	18	DH1	THey	140	32	50	CH	CHurch	150
13	19	IY	sEE	170	33	51	ER1	fIR	110
14	20	EY	bEIge	200	34	52	ER2	fIR	210
15	21	DD1	coulD	50	35	53	OW	bEAU	170
16	22	UW1	tO	60	36	54	DH2	THey	180
17	23	AO	AUght	70	37	55	SS	veSt	60
18	24	AA	hOt	60	38	56	NN2	No	140
19	25	YY2	Yes	130	39	57	HH2	Hoe	130
1A	26	AE	hAt	80	3A	58	OR	stORe	240
1B	27	HH1	He	90	3B	59	AR	alARm	200
1C	28	BB1	Busy	40	3C	60	YR	clEAR	250
1D	29	TH	THin	130	3D	61	GG2	Got	80
1E	30	UH	bOOk	70	3E	62	EL	saddLE	140
1F	31	UW2	fOOd	170	3F	63	BB2	Busy	60

Deze fonemen kunt U rechtstreeks naar de speechunit sturen met P.$28=code#3. Op de plaats van code staat het getal dat U uit bovenstaande tabel kunt kiezen.

Literatuurverwijzing :

AN 3.1	blz 24	F. Cuypers	Speech synthesisers
AN 4.1	blz 14	F. Cuypers	Chatterbox
AN 4.1	blz 76	B. Kasteel	Speech synthesiser
AN 4.3	blz 87	M. v Alphen	Stem aan de Atom
AN 5.1	blz 35	B. Poot e.a.	Project SP0256 al2
AN 5.1	blz 44	B. Poot e.a.	Speech Queue
AN 6.3	blz 37	W. Sobbenmakers	Foneem

Veel plezier ermee ...
... met de vriendelijke groeten van Roland.

```
***************
*   TIMEBOX   *
***************
```

door Roland Leurs

Het programma 'TIMEBOX' dat U op de regioschijf vindt is de source van een real time clock op interruptbasis. Deze klok heb ik wel wat aangepast om wat meer mogelijkheden te krijgen. Een complete gebruiksaanwijzing volgt zo meteen.

Het programma is ongeveer 7 kbytes lang en is goed voor 5 1/2 pagina machinecode. De statements komen niet op #A000, want het geheel is bedoeld als P-Charme uitbreiding. Het is wel mogelijk om het programma onder writeprotect te draaien.

Het programma vraagt na RUN om twee adressen; een voor de tabel met statements en een adres voor de machinecode. Reserveer voor de tabel 1 pagina want achter de tabel wordt ook nog iets geassembleerd.

Er worden enkele zeropage adressen gebruikt door de diverse routines. Hieronder staan ze afgedrukt met hun functie:

0080 : save x register bij printen
0081 : save y register bij printen
023D : uren, tientallen
023E : uren, eenheden
023A : minuten, tientallen
023B : minuten, eenheden
0237 : seconden, tientallen
0238 : seconden, eenheden
0234 : centronics status byte (variabele S)

Het centronics status byte doet al vermoeden dat er iets speciaals aan deze klok is. Het is namelijk mogelijk dat de Atom ons de tijd zegt, door middel van de speechcard aan de centronicspoort. Deze speechboard werkt met de SP0256 AL2 en daar is de software dus ook voor geschreven. Maar als we nu om de tijd vragen kan het natuurlijk ook zijn dat de printer aangesloten is. We moeten dus de computer dus duidelijk maken wie of wat aangesloten is. Dit zou hardwarematig kunnen door de printer een 1 en de speechcard een 0 naar een bepaalde inputpoort te laten sturen, maar dat zou een ingreep in de printer betekenen. Dus doen we dat maar softwarematig. Het resultaat staat dus in #0234.

Ook wordt in de source iets vermeldt van een "break extention byte". Ook dit byte heb ik zelf bedacht. Als mijn Atom zich door de resetroutine heenworstelt dan kijkt hij ook of adres #4FA de waarde #4C bevat. Zo ja, dan volgt een JSR#4FA en dus kan mijn resetroutine uitgebreid worden. Na een break wordt de klok

automatisch opgestart zonder dat de tijd opnieuw moet worden ingesteld. De klok loopt als het ware gewoon door.
Over dit 'break extention byte' zal ook nog wel een artikel verschijnen.

De tijdroutine van dit programma kan natuurlijk ook vervangen worden door een programma dat Big Benny uitleest en dan kan deze software ook voor deze print gebruikt worden. Omdat ik wegens plaatsgebrek in mijn Atom de Big Bennyprint ga verhuizen naar de Electron moet U daar de software zelf voor schrijven.

Genoeg gekletst over de source, nu eens naar de statements. Als we de software gerund hebben, dan saven we de code met de aangegeven adressen. We starten in de toekomst dan op met *RUN "TIME" (of *TIME voor diskgebruikers). De volgende statements zitten in het programma:

```
TIME       - spreekt de tijd uit m.b.v. de SP0256-AL2.
PTIME      - print tijd, gaat verder als print-statement
STIME      - tijd invoeren en break-ext.byte goedzetten
SCREEN x   - zet display; x=0:tijd uit, x=1:tijd aan
SHOW       - geeft informatie en versienummer
p          - zet centronics byte op printer
s          - zet centronics byte op speech
```

Het programma TB-TEST is een testprogramma. Na RUN vraagt het programma waar de timebox geassembleerd is. Voer hier het juiste adres in en het programma laat alle getallen van 0 t/m 59 horen.

Veel plezier ermee ....
.... en de vriendelijke groeten van Roland.

```
>SHOW

TIME BOX V1.10

WRITTEN BY ROLAND LEURS  09.03.89

(c) 1989 SMURFSOFT WATTENHOVEN
(c) 1989 ATOM COMPUTERCLUB LIMBURG

COMMAND'S :
-----------
TIME   - SAY TIME
PTIME  - PRINT TIME
STIME  - SET TIME
SCREEN - TIME OPTION
SHOW   - PRINT INFO
p      - PRINTER CONNECTED
s      - SP0256-AL2 CONNECTED
```

EF9345 VIDEOPROCESSOR

door Peter Wokke

Razend enthousiaste reactie's heb ik gekregen na het verschijnen van het eerste artikel in A.N.7.3 (ze bestaan toch nog die doorgewinterde atomfielen) zodoende hier het vervolg waar ik dieper in ga op het afhandelen van de commando's voor de EF9345. Voor de echte freaken verwijs ik naar de datasheet, (ds 10 is dus datasheet bladzijde 10).

COMMANDO'S: (ds 37)

De 80 kolomskaart heeft 17 hoofdcommando's die afhankelijk van de waarde lezen,schrijven, pointers verhogen, data verzetten enz.,enz.,enz. Niet alle commando's worden gebruikt, dit is afhankelijk van de instelling van de 80 kolomskaart.

INTERN GEHEUGEN: (ds 12)

De processor deelt het geheugen op in districten die weer onderverdeelt zijn in blokken. Dus bij 16Kb ram op de kaart krijg je 4 districten van 4 blokken elk 1Kb groot.

INTERNE COMMANDO'S: (ds 14-16,24,32,36)

ROR
Het ROR commando stelt het district/blok (welke op het scherm moeten komen),en het aantal regels in.

- TGS
Het TGS commando stelt de 40/80 kolomsmode, service line, en het uitgangssignaal in (hor.sync, ver.sync, interlace)

- PAT
Dit commando stelt de atributen die mogelijk zijn in.

- MAT
Dit is wel de meest gebruikte, deze stelt de cursormode in, dus blok, streep, vast, knipperen, aan, uit, en geeft de kleur van de rand aan.

- DOR
Is de meest ingewikkelde met de moeilijkste functie's, want in 40 kolomsmode is het de pointer van het UDS geheugen, maar in 80 kolomsmode stelt het de kleuren van het karakter in.

De inhoud van deze registers worden op adres #BE21 geschreven en/of gelezen, maar het commando wordt gegeven op #BE28.

HOOFD COMMANDO'S: (ds 17-22,30-32,37)

Deze commando's sturen een karakter naar de kaart hierbij moet je rekening houden met 6(5) adressen.

In 40 kolomsmode:		In 80 kolomsmode	
#BE21	karakter	#BE21	karakter
#BE22	atribuut B		
#BE23	atribuut A	#BE23	atribuut(even/odd)
#BE26	regelnummer	#BE26	regelnummer
#BE27	positie op regel	#BE27	positie op regel
#BE28	commando	#BE28	commando

In de 40 kolomsmode worden de karakters en atributen worden apart opgeslagen, dit verklaart dus dat je 3Kb intern geheugen nodig heb voor een scherm. In de 80 kolomsmode gebruik je ook 3Kb voor een scherm maar dan worden de even en de oneven karakters apart opgeslagen en het atribuut samen in een blok.

De adressen #BE24 en #BE25 worden gebruik voor de hulp pointers deze zijn vrijwel gelijk aan de adressen #BE26 en #BE27 en worden alleen gebruikt bij het scrollen.(oude positie en nieuwe positie).
Het laatste adress is #BE20 dit is het zogenaamde status register hierin staan gegevens BUSY,einde regel,enz.
Elk commando is opgebouwd uit twee gedeeltes, de hoge bits geven het commando aan en de lage bits de parameter (lezen/schrijven, pointer verhogen, main pointer, aux. pointer, enz.).

ZEROPAGE GEBRUIK:

Er hoeft geen zeropage gebruikt te worden, maar om toch enige compatibiliteit te verkrijgen zijn de meeste adressen toch in dit gebied geplaatst. Voor uitgebreide opbouw van deze adressen verwijs ik naar txt-3.4. Dit staat op de clubschijf in EDIT80. Dit voor degene die nog geen 80-kolomskaart(????) hebben niet verder te belasten met deze voor enkele nogal saaie kost.

TOEKOMSTSOFT:

Er zal in de toekomst gerust nog wel een nieuwere en/of aangepaste versie van de VDU-soft verschijnen, denk bijvoorbeeld maar aan een versie voor een 65C02 processor, die krachtige instuctie's heeft voor het testen, zetten en resetten van bits , wat veel gebeurd, dus veel sneller afgehandeld kan worden. Helaas is deze processor nog niet in mijn bezit dus kan ik hier (nog) niet aan werken.

OPROEP AAN ALLE ATOMISTEN:

wie heeft een werkeloze 65C02 (of 65C802) processor liggen?
ik ben bereid deze voor een schappelijke prijs over te nemen.

DE BEGINNERSLOGE VAN Roland Leurs

```
BBBBB     AAAAA      SSSS     IIIIII    CCCC
BB  BB   AA   AA    SS  SS      II     CC  CC
BB  BB   AA   AA    SS          II     CC
BBBBB    AAAAAAA     SSSS       II     CC
BB  BB   AA   AA        SS      II     CC
BB  BB   AA   AA    SS  SS      II     CC  CC
BBBBB    AA   AA     SSSS     IIIIII    CCCC
```

deel 6

Nu we ons in de vorige aflevering door de array's hebben geworsteld, komen we vandaag aan bij het laatste deel van het standaard Atom basic gebeuren. In volgende afleveringen komen basic uitbreidingen op papier. In dit deel gaat U het een en ander lezen over tekst.

We hebben al uitgebreid kennis gemaakt met de integer getallen en de floating point getallen. Bij deze getallen kunnen we gebruik maken van constanten en van variabelen. Bijvoorbeeld: constante 4 of 3.1415 ; variabele A B of %C etc.

We kunnen natuurlijk ook met tekst werken. En bij teksten kunnen we ook weer gebruik maken van constanten en van variabelen. De meest eenvoudige manier om een zin op het scherm af te drukken hebben we gezien in de eerste aflevering, nl het PRINT statement. Bijvoorbeeld:

PRINT "ATOM NIEUWS IS EEN GOED BLAD"

geeft na een druk op de <return>-toets :

ATOM NIEUWS IS EEN GOED BLAD

Het gedeelte tussen de aanhalingstekens kunnen we zien als een constante. Er is namelijk geen enkele functie die hier iets aan kan veranderen. In de meeste programma's wordt op deze manier de gebruiker van een programma iets duidelijk gemaakt.

We kunnen bij deze teksten, ookwel strings genoemd, ook gebruik maken van variabelen. Deze variabele is een letter van A t/m Z. De computer weet aan de schrijfwijze van de variabele welke soort bedoeld wordt:

A = integer variabele
%A = floating point variabele
$A = string variabele

De "waarde" van een string variabele moet natuurlijk ook ergens in het geheugen komen te staan, net zo als de gewone en fp variabele. Deze plaats maken we vrij met het statement DIM. Dit statement hebben we ook al gezien bij de array's.

Met de instruktie DIM A(63) reserveren we 64 geheugenplaatsen voor de string A. De variabele A kan nu max. 64 tekens bevatten maar minder mag ook.

Bijvoorbeeld:

```
10 DIM A(63)
20 $A="** DIT IS NU STRING A **"
30 PRINT $A '
40 END
```

geeft na RUN :

```
** DIT IS NU STRING A **
```

Zo eenvoudig is de eerste stap met strings. Er zijn nog meer mogelijkheden. Het is ook mogelijk om strings in te voeren m.b.v. het INPUT statement:

```
10 DIM A(63)
20 INPUT "VOER UW TEKST IN ",$A
30 PRINT "U TYPTE : ",$A'
40 END
```

Als U dit programma intikt en RUNt ziet U wat er gebeurd. Als U nu meer dan 64 tekens invoert volgt er geen foutmelding. Toch kan er iets fout gaan als U meerdere strings gebruikt. Typ onderstaand voorbeeld maar eens in:

```
10 DIM A(10),B(63)
20 $A="DEZE STRING IS DUIDELIJK TE LANG"
30 $B="** DIT IS NU STRING B **"
40 PRINT $A',$B'
50 END
```

En U kunt na RUN zien dat $A niet meer klopt. Dus moeten we de Atom op de lengte laten letten. Hoe dit in zijn werk gaat volgt verderop in dit verhaal.

De Atom biedt ons ook nog enkele functies m.b.t. strings. We kunnen strings zelfs vergelijken met elkaar en bij elkaar optellen. Voordat ik U laat zien hoe we strings moeten vergelijken is het handig dat U iets meer weet omtrent de tekens op het toetsenbord.

Als we op een toets drukken, dan krijgt de Atom van het "toetsenbord-ic" een bepaalde code aangeboden. Deze code wordt in de ROM omgezet naar een zgn ASCII-code. Een tabel van deze code vindt U op de achterkant van het Error Code kaartje. Ook staat een dergelijke tabel op blz 152 van de Nederlandse Atom handleiding van P.J. Smulders.

Ieder karakter heeft een specifieke ASCII code, bijvoorbeeld de code voor de letter A is 65. Een * heeft als code 42. Dit kunt U zien met deze instruktie: PRINT $65 geeft een A, PRINT $42 geeft een *.
We kunnen van ieder karakter in basic de ASCII waarde opvragen met de functie CH. Dus PRINT CH"A" geeft 65 en PRINT CH"*" geeft 43, PRINT $CH"A" geeft weer een A.
Als we nu twee karakters gaan vergelijken dan vergelijken we in feite dus de ASCII waarde want je kunt niet zeggen dat * groter of kleiner is dan A; maar wel dat 42 kleiner is dan 65.

Met deze kennis in het achterhoofd kunnen we gaan vergelijken. Op de gebruikelijke manier staat de Atom alleen het = teken toe. Bijvoorbeeld:

```
10 DIM I(2)
20 PRINT "HOEVEEL STERREN WILT U ZIEN ",S
30 FOR X=1 TO S
40   PRINT $42   ;REM CODE VOOR EEN STERRETJE
50 NEXT X
60 INPUT '"WIL U NOG MEER STERREN ZIEN [ja OF nee] "$I
70 IF $I="JA" OR $I="ja" THEN GOTO 20
80 IF $I="NEE" OR $I="nee" THEN GOTO 90
85 PRINT "IK VWORG ! ";GOTO 60
90 PRINT "TOT ZIENS"'
100 END
```

Het voorbeeldprogramma vraagt dus eerst hoeveel sterren U wil zien. Vervolgens wordt het gewenste aantal sterretjes afgedrukt. Daarna verschijnt de vraag of U nog meer sterren wil zien. U kunt hier antwoorden met JA, ja, NEE of nee. Als U ja antwoordt dan wordt het programma vanaf regel 20 weer doorlopen. Als U met nee antwoordt dan stopt het programma. Als U een onzinnig antwoord geeft wordt de vraag opnieuw gesteld.

De andere vormen van vergelijken "<" en ">" kunnen we niet direct toepassen. Om bijvoorbeeld strings te sorteren moeten we gebruik maken van een subroutine. In deze subroutine kunnen we de ? operator gebruiken om de letters een voor een te lezen en te vergelijken:

```
1000 REM VERGELIJK STRINGS
1010 REM VERGELEKEN WORDEN DE STRINGS A EN B
1020 IF $A=$B THEN PRINT "GELIJK"
1030 X=-1
1040 DO X=X+1
1050 IF X?A < X?B THEN P."$A IS KLEINER DAN $B";X=-1;G.1070
1060 IF X?A > X?B THEN P."$B IS GROTER DAN $B";X=-1
1070 UNTIL X?A=13 OR X?B=13 OR X=-1
1080 RETURN
```

Als het programma in deze subroutine komt wordt eerst vergeleken of de strings hetzelfde zijn. (r.1020) Zo ja, dan terug via RETURN. In regel 1030 wordt een teller ge-initialiseerd. In een DO-UNTIL lus wordt de teller verhoogd en worden de tekens in de strings vergeleken.

A	C	O	R	N	~								= $A
A	T	O	M	~									= $B
↑ X													

In bovenstaande tabel staan schematisch twee strings weergegeven. $A="ACORN" en $B="ATOM". X wordt als teller gebruikt.
Als X=0 staat de teller bij het begin van de string. In regel 1050 kijkt het programma of de inhoud van adres X+A (dus A als X=0) kleiner is als de inhoud van adres X+B. In dit geval zijn ze gelijk dus de vergelijking is onjuist. Het programma gaat verder op regel 1060. Hier wordt gekeken of de inhoud van X+A groter is dan de inhoud van X+B. Ook niet, dus verder bij regel 1070.
Hier wordt gekeken of een van de twee adressen de code 13 bevat. Deze code is de stopcode van een string. (In de tabel met ~ weergegeven). Zo ja, dan is de vergelijking afgelopen want de code 13 is kleiner dan de kleinste code voor een teken. Dat is de spatie met de code 32.
Ook wordt gekeken of het controle-getal juist is. Dit getal wordt gezet als een vergelijking klopt. Hierdoor wordt de subroutine afgebroken want er is een resultaat.
Als er nog geen 13 staat en als het controle getal nog niet klopt wordt de teller verhoogd. De X wordt 1 en komt dan in de tweede kolom te staan. Hier staan een "C" resp. "T". Als U bovenstaand verhaal weer herhaald, merkt U dat de "C" kleiner is dan de "T" en dus verschijnt de melding "$A<$B". De subroutine wordt verlaten en het programma gaat zijn weg verder.
Deze subroutine kan gebruikt worden voor het sorteren op alfabetische volgorde van strings.

Zoals eerder gezegd in het verhaal kunnen we de Atom de lengte laten bepalen van een string. Hiervoor hebben we de functie LEN. Bijvoorbeeld :

```
10 DIM A(64)
20 $A="BASIC DEEL 6"
30 PRINT $A'"TOTAAL : ",LEN(A)", TEKENS"
40 END
```

geeft na RUN :

```
BASIC DEEL 6
TOTAAL :       12 TEKENS
```

immers spaties tellen ook mee !

Op deze manier kunnen we er dus voor zorgen dat een string niet te lang worden bij een invoer:

```
10 DIM F(7)
20 INPUT "FILENAAN "$F
30 IF LEN(F) >7 THEN PRINT "NAAM TE LANG"'
40 END
```

Ook kunnen we met de functie LEN een string een vaste lengte geven:

```
10 DIM F(13)
20 INPUT "GEEF EEN WOORD ;"$F
30 IF LEN (F) < 13 THEN DO $F+LEN F=".";UNTIL LEN (F)=13
40 PRINT $F'
50 GOTO 20
```

Hier wordt bij de bestaande string net zovaak een "." toegevoegd totdat de lengte 13 is. Dan wordt de string ter controle afgedrukt.

We hebben in dit laatste voorbeeld dus ook al gezien hoe we strings moeten optellen. In de algemene vorm gaat dit als volgt:

$A+LEN(A)=$B

Hier tellen we $B op bij $A. In de praktijk komt het erop neer dat $B direct achter $A geplaatst wordt.

```
10 DIM A(10),B(5)
20 $A="ACORN"
30 $B="-ATOM"
40 $A+LEN(A)=$B
50 PRINT $A'
60 END
```

De uitvoer kunt U zich wel indenken: ACORN-ATOM.

We kunnen strings niet van/met elkaar aftrekken/vermenigvuldigen of delen. Wel kunnen we strings korter maken. Hiervoor hebben vele computers standaard functies: MID$, RIGHT$ en LEFT$. Onze Atom kent deze functies niet. Maar wij basicprogrammeurs vinden hiervoor natuurlijk toch een oplossing. Hieronder zal ik deze functies een voor een uitwerken:

De functie LEFT$:

Deze functie geeft het linkergedeelte van de string. Het aantal karakters bepaalt de programmeur (u dus). Dit kan heel eenvoudig. Stel $A="ACORN ATOM" en we willen in $B de eerste 5 letters uit $A plaatsen. Dan maken we eerst $B gelijk aan $A. Vervolgens plaatsen we op de N-de plaats een return:

```
10 DIM A(10),B(10)
20 $A="ACORN ATOM"
30 $B=$A          ; REM COPIEER $A
40 $B+5=""        ; REM PLAATS EEN RETURN-CODE
50 PRINT $B       ; REM EN HET ANTWOORD IS 'ACORN'
60 END
```

De functie RIGHT$:
Deze functie geeft het laatste gedeelte van een string. Beginnend bij het N-de karakter. Als we in $B de laatste vier letters willen plaatsen van $A dan copieren we dus alleen maar de laatste vier letters:

```
10 DIM A(10),B(10)
20 $A="ACORN ATOM"
30  $B=$A+6        ; REM WE COPIEREN VAN  $A ALLES ACHTER HET 6e
KARAKTER
40 PRINT $B       ; REM DUS HET WOORDJE 'ATOM' WORDT AFGEDRUKT
50 END
```

De functie MID$:
Deze functie geeft een middenstuk van een string. Het begint bij het N-de karakter en eindigt bij het N-de karakter. Het ligt voor de hand dat deze functie een combinatie is van de twee bovenstaande functies:

```
10 DIM A(30),B(10)
20 $A="ACORN ATOM MICROCOMPUTER"
30 $B=$A          ; REM WEER COPIEREN
40 $B+10=""       ; REM HET LINKERGEDEELTE EERST
50 $B=$B+6        ; REM HET RECHTERGEDEELTE DAARNA
60 PRINT $B       ; REM ACHTERHAAL ZELF DE UITKOMST ! ! !
70 END
```

Dit is een omslachtige manier om deze functies te verwezenlijken maar het werkt wel. Op de volgende bladzijde staan de zojuist behandelde functies eens in P-Charme Basic.
Met P-Charme in onze Atom kunnen we zelf functies en procedures maken. Procedures kunnen we beschouwen als statements binnen het programma. Met het statement FUNCTION maken we onze eigen functie's. Dit hoeft U (nog) niet te begrijpen. Het is meer ter illustratie bedoeld.

```
FUNCTION LEFT(S,L)
  IF L<1 OR LEN(S)>64 THEN ERROR
  $#140=$S;L?#141=13
  LEFT=#140
FEND

FUNCTION RIGHT(S,L)
  IF L<1 OR LEN(S)>64 THEN ERROR
  $#140=$S;IF L<LEN(S) THEN $#140=$S+LEN(S)-L
  RIGHT=#140
FEND

FUNCTION MID(S,L,M)
  IF L<1 OR M<1 OR LEN(S)>64 THEN ERROR
  ?#140=13;IF L<LEN(S) THEN $#140=$S+L-1;M?#140=13
  MID=#140
FEND
```

Gebruik:

```
$S="THIS IS A STRING"
PRINT $LEFT(S,6)'          ... THIS IS
PRINT $RIGHT(S,10)'        ... S A STRING
PRINT $MID(S,2,6)'         ... HIS IS
$A=$LEFT(S,5)
```

Zo krijgt u enige indruk wat mogelijk is met de P-Charme uitbreiding. Want zeg nou eerlijk, dit is veel duidelijker dan de eerste methoden.
Overigens biedt P-Charme ons nog een string-functie : INSTR.
Met behulp van deze functie kan gekeken worden of een bepaalde string in een andere string voorkomt. De syntax is: INSTR(string1,string2).
De functie onderzoekt of string2 in string1 voorkomt. Zo ja, krijgt INSTR de waarde van de positie van die string in de eerste string. Zo nee, dan is de waarde '0'. INSTR kan ook gebruikt worden bij het IF-statement en andere logische expressies. Enkele voorbeelden:

```
P=INSTR($A,"HALLO")
PRINT INSTR($B+6,$C); REM ZOEKT VANAF DE 6e POSITIE IN $B
IF INSTR($D,$B) THEN PRINT "GEVONDEN !"
```

Hier wou ik het bij laten voor deze keer. Vanaf de volgende aflevering zal ik verder gaan met statements uit de meest gebruikte uitbreidingsroms. Ik zal niet uitgebreid op alle statements ingaan want dan is het einde zoek. Waarschijnlijk staat dan de laatste Atom in het Rijksmuseum als ik stop met deze serie. Zoveel is er immers al gemaakt! Wat er precies gaat gebeuren leest u de volgende keer.

Mocht U nog vragen hebben bel of schrijf gerust, mijn naam is vast wel ergens te vinden. Tot een volgende keer en natuurlijk:

... met de vriendelijke groeten van Roland ...

```
**********************
* SOURCE PROGRAMMA'S *
**********************
```

door G.J. van den Brandhof

Source van DB3, Database voor de Atom

Op de regioschijf vindt U drie programma's die samen de source vormen van DB3-DOS. Deze programma's moeten in onderstaande volgorde gerund worden:

1. DB3-ST
2. DB3-1
3. DB3-2 (Let op, Ram op #A000)
4. DB3-1 (tweede slag)

Dit alles omdat de source te lang is !

Er is nog enige onduidelijkheid omtrend de labels NN18 t/m 25. In de oorspronkelijke source zitten fouten waar naar deze labels wordt verwezen. Echter de database loopt er in ieder geval niet op vast.

Voor een duidelijke handleiding verwijs ik naar Atom Nieuws jaargang 4, nummer 1, blz 48 t/m 57.

Source van CALCROM

Verder vindt u op de regio schijf ook nog de source code van de bekende "CALCROM" van Acornsoft. Deze source heeft verder geen toelichting nodig.

Vraag van de redactie

Nu de source van zowel Dbase3 als Calcrom beschikbaar zijn kunnen deze programma's aangepast worden voor de 80 kolomskaart. Wie grijpt hier de kans om aan zijn levenswerk te beginnen?

```
*************************
*   TABEL VOOR P-CHARME  *
*************************
```

Het programma DIR.AT# is een verbeterde versie, die al eerder in ATOM NIEUWS heeft gestaan (Jrg.5 nummer 4 blz. 27), vooral daar het oude programma geen rekening hield met meerdere statement/commando lijsten, dit programma doet dit wel.

Dick Protzman

```
>L.
   10 PROGRAM TABEL
   20 REM  ppr ;(c)DICK PROTZMAN '88
   30 REM ONDER DEZE REGEL KUNT U HET PROGRAMMA WEIZIGEN
   40 DIM N20;Q=0;P.$12
   50 HTAB 5:P."DIT PROGRAMMA MAAKT"
   60 P." EEN TABEL AAN VOOR P-CHARME"'
   70 HTAB 5;P."BIJ HET LABEL a KUNT U UW"
   80 P." ASSEMBLER CODE KWIJT"''
   90 HTAB 5;P."LAATSTE TABEL ADRES #",&!#291A'
  100 HTAB 5;P."LAATSTE M.C. ADRES  #",&!#291E''
  110 HTAB 5;P."HEBT U AL EEN TABEL VAN EEN BESTAANDE"'
  120 HTAB 5;P."UITBREIDING ZOALS 'ACE' OF"'
  130 HTAB 5;P."EEN EIGEN ONTWERP  [Y/N] ;?"
  140 DO INKEY Y;U.Y=CH"Y"OR Y=CH"N";P.$Y''
  150 XIF Y=CH"Y" THEN
  160 HTAB 5;P."WAAR STAAT DEZE  [MET #] :       "$8$8$8$8$8$8;IN.E
  170 IF E%256<>0;P.$7$11;G.160
  180 ELSE
  190 HTAB 5:P."UW TABEL OF NIEUW STATEMENT"'
  200 HTAB 5;IN."MOET  KOMEN  OP  [MET #] :"T
  210 XIF T%256=0 AND Y=CH"Y" THEN P.'
  220 HTAB 5;P."U KUNT DE NU AANTEMAKEN TABEL WEER VERWIJDEREN"'
  230 HTAB 5;P."DOOR HET EERSTE ADRES #FF TE MAKEN DUS ?#",&E,"=#FF"'
  240 HTAB 5;P."NIET VERGETEN - e.v.t WRITE PROTECT UITSCHAKELEN !!"''
  250 ELSE
  260 HTAB 5;IN."UW CODE  MOET OP [MET #] :"A
  270 G=T;P.'
  280 HTAB 5;IN."NAAM VAN UW STATEMENT IS :"$N
  290 P.$12'
  300
  310 REM ASSEMBLEREN VAN HET PROGRAMMA
  320 ASM-VERSION;P."IS ASSEMBLING YOUR PROGRAM"''
  330 HTAB 5;P."PASS 1"';PASS 2;GOS.a
  340 HTAB 5:P."PASS 2"';PASS 1;GOS.a
  350 ASM-WARNING;P=P&#0000FFFF;P.'
  360 HTAB 5;P."CODE FROM #"&A" TO #"&P'
  370
  380 REM MAAK DE TABEL
```

```
390 XIF T%256=0;T!0=#00C6E3FF;T=T+3
400 ELSE
410 $T=$N;T=T+L.N
420 T?0=(start'asm)/256;#80
430 T?1=(start'asm)%256;T?2=#80
440 XIF G%256=0 THEN
450 XIF Y=CH"Y" THEN ?E=G/256
460 ELSE ?#3FC=G/256
470 ELSE
480 T=T+2;!#29|A=T;!#29|E=P
490 HTAB 5;P."NEXT TABLE ADRES IS #"&T'
500 HTAB 5;P."ALS U NU DIT PROGRAMMA SAVED,"'
510 HTAB 5;P."SLAAT U DE LAATSTE TABEL EN M.C."'
520 HTAB 5;P."ADRESSEN MEE OP"
530 P.''"EINDE"';END
540
550*ASM-BEGIN
560
570.OPTION #40
580.TABLE #9800,#A000
590
600\\              START ZEROPAGE GEBIED
610:zp=#70
620
630:esc=#1B
640:del=#7F
650:bel=7
660:bs=8
670:ht=9
680:lf=10
690:vt=11
700
710.CODE A
720
730\\              DIR AT # - P-CHARME BOXEN
740:start'asm JSR#C4E4
750 LDA#3FC;LDY@0;STY zp
760:label'0 STA zp+1;INY;LDA(zp),Y
770 CMP@#E3;BNE label'5
780 INY;LDA(zp),Y;CMP@#C6;BNE label'5
790 JSR#FFED;JSR#F7D1
800.ASCII"  DIR AT #".#EA
810 LDA zp+1;JSR#F802;JSR#CD54;LDY@3
820:label'1 LDA(zp),Y;BMI label'2
830 JSR#CA4C;INY;BNE label'1
840:label'2 CMP@#80;BEQ label'4
850 EOR@#80;PHA
860:label'3 JSR#F879;LDA#7;CMP@10;BNE label'3
870 PLA;JSR#F802;INY;LDA(zp),Y
880 JSR#F802;JSR#CD54;JSR#C504;INY;BNE label'1
890:label'4 LDY@0;LDA(zp),Y;CMP@#FF;BNE label'0
900:label'5 JSR#FFED;JSR#C55B
910
920\\ #FFED - GEEF EEN RETURN - CODE'S 13,10
930\\ #F7D1 - PRINT EEN STRING - #EA=NOP INSTUKTIE
940\\ #F802 - PRINT EEN BYTE IN 2 # GETALLEN
```

```
 950\\ #CD54 - RETURN EN #7 OP 0
 960\\ #CA4C - PRINT KARAKTER EN VERHOOG #7
 970\\ #F379 - PRINT EEN SPATIE EN VERHOOG #7
 980\\ #C504 - KIJKT NAAR ESC TOETS
 990\\ #C55B - TERUG NAAR INTERPRETER
1000
1010.END OF ASSEMBLY
1020 RETURN
```

Beter laat dan nooit ...

Sommige mensen vallen altijd over de gekste fouten. Zo wees Chris Hilkman uit Urmond mij op een maandagmorgen op een fout in het artikel 'verborgen 6502 instruktie's' van Rudi van Drunen. Dit artikel behandelt enkele opcodes die de assembler niet herkent, maar die wel iets doen. Overigens verscheen dit artikel in Atom Nieuws jaargang 3 (1984!) nummer 6.

Waar draait het nu om? Wel kijk, dit staat letterlijk in de tekst:

SAX : AND Accu with x reg. and store result.

flags: N Z C I D V
X X - - - -
zero: 78 abs: 8F abs,x: 9F abs,y: 97

NB this instruction takes 3 extra cycles

De fout zit in de opcode 78. Dit is de instructie SEI. Dus dat is fout! De juiste opcode voor deze instructie is 87. Je hoeft het maar te ontdekken...

Met de vriendelijke groeten van Roland.

SIE PIE EMMERTJES

Voor diegenen die niet zo vertrouwd zijn met CPM zal de opstart van de eerste CPM-schijf als normaal worden aangenomen.
Bij een koude start (reset) verschijnt er op het scherm de kreet 'STARTU?'.
Dit is een foutmelding onder CPM, er werd een file gezocht onder de naam 'STARTUP', en deze werd niet gevonden.
Startup is een bootfile welke automatisch bij een koude start de desgewenste kommando's uitvoert.
De foutmelding wordt overigens veroorzaakt door het feit dat de file abusievelijk onder de naam 'STARTU' is gesaved.
Deze file is trouwens door de maker specifiek voor hem geschreven.
Het herstel van deze fout gaat als volgt:

DIRECT MODE RUN OF STARTUP.

```
A>STARTU S
STARTUP, Version 1.4
STARTUP  Setup command (?=Help)? C
         Current Multiple Command Buffer Contents is --
R PRKEY  PASCAL;R TIME 0:0:0
         New Line? DIR
         Multiple Command Line --
         --->DIR
STARTUP  Setup Command (?HELP)? X
         Do you wish to update STARTUP (Y/N/<CR>=Y)? Y
         STARTUP will write file to current disk/user
         STARTUP File is STARTUP.COM
         Enter New File Name or <CR> if OK:
         STARTUP File is STARTUP.COM
         Enter New File Name or <CR> if OK:
         Writing File STARTUP.COM to Disk ... Done
```

Je hebt nu een STARTUP.COM file gekreeerd die bij een koude start er voor zorg draagt dat er automatisch een 'DIR' van de desbetreffende diskette wordt gegeven.
Deze file is uiteraard door middel van NSWEEP op al je diskettes te kopiëren.
Voor een uitbreiding van de MULTIPLE COMMAND BUFFER raad ik je aan om de file Z2CON.WS te bekijken (b.v. m.b.v. TYPE).

Wil Kautz

Padding Programma .

B.Geene.

Dit programma berekent de zelfinducties en capaciteiten van spoelen en condensatoren van antenne en oscillator-kringen in een superheterodyne radio-ontvanger,zowel voor het AM- als het FM-bereik.
Ook wordt de gelijkloop-afwijking (PADDING) tussen antenne- en oscillatorkringen berekend.
In de wat goedkopere radio-ontvangers wordt de zenderkeuze bijna altijd gerealiseerd met behulp van een variabele condensator, kortweg genaamd "varicon".
In het bovengenoemde type ontvanger wordt het uit de lucht opgepikte zendersignaal met een bepaalde frequentie door de antenne-kring uitgefilterd en samen met het in de ontvanger opgewekte oscillator-signaal aan de zogenaamde menger toegevoerd.
Deze menger maakt dan van deze twee signalen een heleboel andere signalen met de som-,verschil- en productfrequenties,waarvan in dit geval alleen de verschilfrequentie door het middenfrequent-filter wordt uitgefilterd,om verder in de radio te worden versterkt en bewerkt.
In formule : Fosc-Fant = Fmid. Zie fig 1.
In de figuren zijn zoveel mogelijk de variabelen uit het programma aangegeven.
Omdat een radio ,waarmee maar een zender kan worden ontvangen , niet erg practisch is ,worden de frequentie-bepalende kringen variabel gemaakt met de varicon .
Dit is een condensator met twee of meer stel verdraaibare platen, zodanig,dat de capaciteit varieert tussen b.v. 5 en 145 pf.
De delta-capaciteit is dan 140 pf.
Met deze delta kunnen we dan een frequentie-verhouding maken van bijv.1 : 3.
Maar,omdat de oscillator-frequentie een constante waarde Fmid hoger is dan de antenne-frequentie,is de verhouding :
Fosc.laag/Fosc.hoog kleiner,
Dus de delta-capaciteit van de oscillator-kring zal kleiner moeten zijn dan die van de antennekring.
Dit kan op twee manieren worden opgelost.
Ten eerste:door ongelijke condensatoren te nemen,bijv.140/82 pf.
Ten tweede:door een serie-condensator toe te passen.
De vervang-capaciteit is dan kleiner,waardoor de verhouding Fosc.laag/Fosc.hoog ook kleiner wordt.
Een en ander is getracht duidelijk te maken in fig.2.
Lijn 1 geeft het frequentie-verloop van de antenne-kring als functie van de draaihoek I van de varicon.
Lijn 2 geeft het frequentie-verloop van de oscillator-kring , zoals het zou moeten zijn; n.l. Fmid hoger dan de antenne-kring.

Daar de middenfrequentie een vaste waarde Fmid moet hebben,zien we dat een van de bovengenoemde correcties moet worden toegepast; we krijgen dan het verloop volgens lijn 4.
Het verloop kan nog verder worden verbeterd,door de nulpunten voor Fmid niet aan de uiteinden,maar iets vanaf die uiteinden te leggen; we krijgen dan de lijn 5.
Het is te zien dat er dan nog + en - afwijkingen t.o.v. Fmid overblijven.
Deze afwijking is de zogenaamde PADDING .
Het programma berekent deze padding.
In feite is dat (lijn 5) - (lijn 3).
De figuren 3 en 4 geven de circuits waarmee gerekend wordt.
Nog enkele practische waarden :

In de DATA-statements zijn waarden opgenomen van enkele zeer gangbare varicons (10*10 graden)
Voor AM zijn dat 140/82 pf en 2* 235 pf; voor FM 2*20 en 2*40 pf
Gangbare middenfrequenties voor AM zijn 450,455,460,en 468 kHz ; voor FM is dat 10,7 MHz.
Gangbare frequentie-bereiken voor AM zijn :

Lange golf	LW	147-275 kHz,
Middengolf	MW	512-1635 kHz,
Korte golf	SW1	2.25-7.45 MHz
	SW2	9.3-22.29 MHz,
SW1 europa		5.82-18.2 MHz,
SW2 europa		3.87-12.3 MHz,
En voor FM		87.35-108.2 MHz.

De software.

Het programma is zodanig opgezet dat eerst gevraagd wordt naar wat men wil ,d.w.z. welk bereik en welke varicon men heeft.
Dan de eerste berekening en daarna kan men aan de hand van de eerste resultaten allerlei variaties uitvoeren met frequenties en seriecondensatoren.
Men kan dus trimmen naar het goede resultaat.
Vooral de padding kan heel goed geoptimaliseerd worden ,door de serie-condensator steeds een andere waarde te geven,naar gelang het resultaat.
Dat is optimaal ,als de waarde in het midden door nul gaat.
Bij de start van het programma wordt voor de serie-condensator in twee gevallen al een waarde ingevuld, afhankelijk van de keuze van bereik en varicon.
De listing is vrij duidelijk zou ik zeggen,omdat geen afkortingen zijn gebruikt ,met sub-routines is gewerkt en nogal wat REM-statements zijn toegevoegd.
Daarom is de lengte 4.5 k geworden.
"IF" staat voor "intermediate frequency" ,dus middenfrequentie, "RF" staat voor "radio frequency",dus ontvang-frequentie.
De floating-point-notatie suggereert een grote nauwkeurigheid.
Dit is niet het geval, daar de spoelen en de varicon een zekere paracitaire capaciteit hebben,die in de berekende waarden van XC en XX zijn begrepen.
De "echte" waarde van XC en XX zal ongeveer 5pf minder zijn .

Sjaak Geene
Zonnewijde 6,
's Bosch .

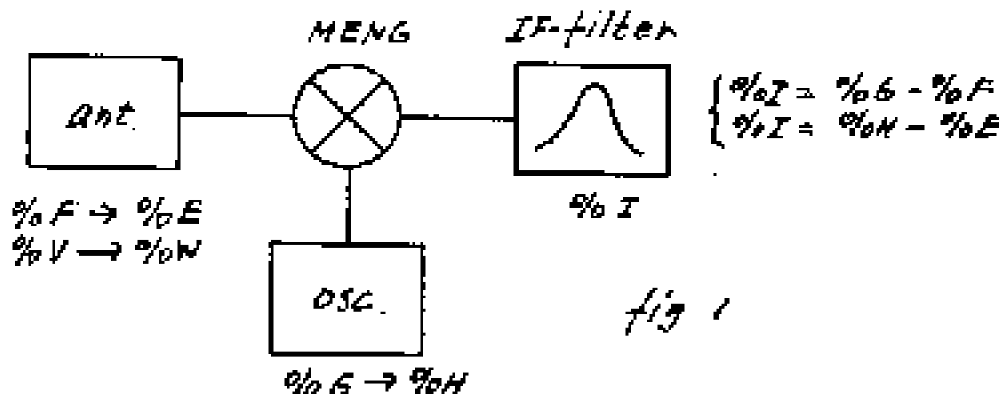


fig 1

I °
100
90
0
1
2
3
4
5
f uid.
500 RF. laag %F %V
1000 OSC. laag %G
1500 RF. hoog %E %W
2000 OSC. hoog %H
2500
3000
→ f (kHz)

fig 2

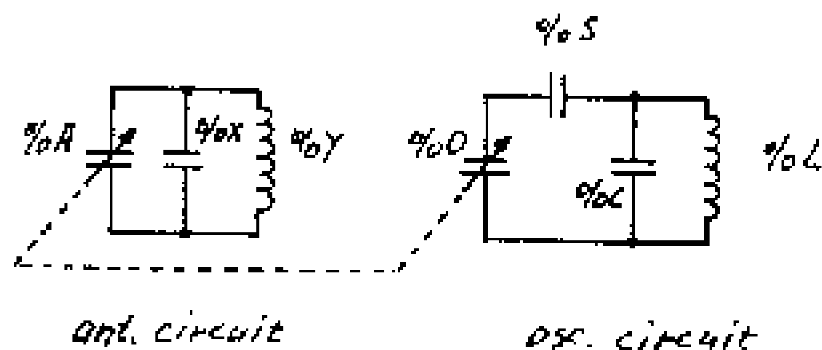


fig 3

DISKCAT VOOR GDOS-SCHIJVEN OP "Z-80-SYSTEMEN"

B.Meintema.

Een rectificatie en een aanvulling van het artikel in AN 8-1 p.78

Er zullen hoogst waarschijnlijk mensen zijn die bij het gebruik van het door mij (aan Z80-systemen) aangepaste DISKCAT ,een "error" te zien kregen.Hiervoor mijn verontschuldiging.
Ik had namelijk een,specifiek aan mijn eigen systeem aangepaste versie opgestuurd ;hierbij dan de algemeen bruikbare.
Deze versie is bovendien ten opzichte van de vorige eenvoudiger in hantering ,-en-,nu ook toepasbaar op GDOS-schijven.
In de normale hantering zijn nog slechts 2 files in gebruik :
 * DCATado en FCAT (voor ATOM) resp. DCATgdo en FCAT (voor GDOS).
Indien men van gemak houdt kan de file "L"gebruikt worden voor opstart bij normaal gebruik.
Voor een allereerste opstart moeten de files "startad",resp "startgd" gebruikt worden om de desbetreffende (lege)FCAT-file aan te maken.
De ATOMDOS-versie is geschikt voor 79 schijven, de GDOS-versie kan in totaal 34 schijven bevatten.
Mensen zonder Z80-kaart kunnen deze programmatuur ook gebruiken, maar moeten LINK#ECA6 veranderen in LINK#EE15 voor GDOS.
Voor het aanpassen van de printercodes :
   ADOS regels 550,610,810,1150,1190,1250
   GDOS regels 810,815,818,840,845,850,855,910
Voor aanpassing van de VDU statements :
   ADOS regels 100,310,390,400,630,830
   GDOS regels 70,80,250,260,270,290,460,1170,1000
DCAT.. wordt geladen van #8200 tot # 9500,voor FCAT is beschikbaar #2900 tot #8200.
Zie verder de,in het vorige artikel genoemde beschrijvingen.

vriendelijke groeten ,B.MEINTEMA

OUD AMBACHT 131
9201VP DRACHTEN.

Meer SPACER's -versie 2 en 3.

H.Bastings.

Na het aanpakken van enige andere programma's met behulp van m'n spacer ,kwam ik tot de ontdekking ,dat er toch nog enkele verbeteringen mogelijk zijn.
Naar mijn eigen oordeel is dit aardig gelukt, zodat, met enig om-schrijven ,het geheel nog steeds binnen 256 bytes past.

NADELEN SPACER VERSIE 0.

Na het gebruik van SPACER versie 0 ,kan het gebeuren dat het programma regels mist, als je na het gebruik van de "unspace"-optie weer direct gaat editten.
Wat gebeurt er dan eigenlijk.?
Een regel die alleen uit een spatie bestaat ,zal na het gebruik van de "unspace"-optie,alleen nog uit een regelnummer bestaan.
Ga je nu het programma wijzigen,of wis je een lege regel,dan wordt tevens de volgende regel uit jouw programma verwijderd.
Zijn er al programma's met SPACER versie 0 bewerkt,dan is deze fout alleen te herstellen door achter het desbetreffende regel-nummer een spatie te plaatsen,en de volgende regel met de copy-toets te copieren.

Een andere mogelijkheid is ,met SPACER 2 of 3 , een spatie toe te voegen , en vervolgens weer te "unspacen ".

Nu iets over de nieuwe versie's.

VERSIE 2.

Versie 2 is alleen bedoeld voor BASIC programma's.

De volgende wijzigingen zijn doorgevoerd t.o.v. versie 0 :

1. versie 2 laat bij een regel die alleen uit spatie's bestaat, een spatie staan.
2. aan het begin van een regel worden alle spatie's verwijderd.

VERSIE 3.

Versie 3 is aangepast aan het gebruik bij assembler-programma's.
Achter de regelnummer blijven staan :

1. "de punt",zoals gebruikt in symbolische assembler,bv. ".code"
2. de dubbele punt (:),gebruik bekend,
3. de " \ " voor commentaar,
4. het openings - en sluitingsteken ([en]),
5. de tekens a t/m z (basic -labels)

Hier dan de listings.

Listing SPACER versie 2.:

```
0REM SPACER2
1P.$12;IN."SPACE 1 UNSPACE 0 "A
2IN."FROM "P;IN."TO    "E
3IF?P=13;IFP?1=#FF;!E=!P;P."ready"'';E.
4!E=!P;E=E+3;P=P+3
5IFA=0;G.9
6?E=32;E=E+1
7IF?P>95;IF?P<123;E=E-1
8G.10
9IF?P=32;IFP?1<>13;P=P+1;G.9
10$E=$P;E=E+L.E;P=P+L.P;G.3
```

Listing SPACER versie 3.:

```
1P$12;IN."SPACE 1 UNSPACE 0"A
2IN."FROM "P;IN."TO    "E
3IF?P=13;IFP?1=#FF;!E=!P;P."ready"'';E.
4!E=!P;E=E+3;P=P+3;IFA=0 G.8
5?E=32;E=E+1
6IF?P=46OR?P=58OR?P>90;IF?P<123;E=E-1
7G.9
8IF?P=32;IFP?1<>13;P=P+1;G.8
9$E=$P;E=E+L.E;P=P+L.P;G.3
```

Waarschuwing :
Wil je beide versies binnen 256 bytes houden,dan mogen ze niet worden hernummerd !.

Het gebruik van de spacers.

Nog even iets over het gebruik van de spacer's.
Deze programma's dienen ter verbetering van de leesbaarheid van bestaande programma's.
Wil je een programma van spaties voorzien,dan laad je een van beide versies op bijv. #2800 (in dat geval mag je vanaf #2900 het te bewerken programma laden.)
Na het starten van spacer ,vraagt deze waar het te bewerken programma begint ,doormiddel van de vraag "FROM".
Je zou dus kunnen antwoorden met #2900.
Het programma mag echter ook overal elders in het geheugen staan.
Op de vraag "TO" geef je het beginpunt op ,vanaf waar het herziene programma wordt neergezet .
Bijvoorbeeld : TO ? #3000.
Denk eraan dat voor het werken met spacer 2* zoveel geheugen (minimaal) nodig is ,als het te bewerken programma lang is.

Ondanks het feit ,dat SPACER in Basic is geschreven,is dit programma verassend snel.

Veel succes bij eventueel gebruik.
H.Bastings
Regio Limburg.

```
*********************************
* E L E C T R O N   N I E U W S *
*********************************
```

Een joystick aan de Electron

naar een idee van Rob Bronsveld

Als je op je Electron veel spellen speelt verlang je soms wel naar de Atom. Die heeft tenminste een joystick. Natuurlijk kun je voor de Electron ook wel zo'n ding halen, maar er komt bogal wat meer bij kijken: om te beginnen heb je een plus-1 nodig. In die plus-1 steek je een analoge joystick (en dat vind ik rotdingen) of een digitaal joystick interface met daaraan een digitale joystick. En als je dat allemaal hebt ga je Hopper spelen en merk je dat dat spel niet werkt met joystick. Wat een enorme ellende.

Rob die bedacht het uitstekende idee om een joystick aan te sluiten op de wijze zoals we dat bij de Atom gedaan hebben. Echter zo gemakkelijk als bij de Atom was dat niet omdat de toetsenbord structuur helemaal anders is (het toetsenbord is namelijk een van de "sideway-roms" en wel romnummer 9). Het toetsenbord wordt net als een geheugen ic geadresseerd en uitgelezen. Dus kwam er wat extra hardware aan te pas.

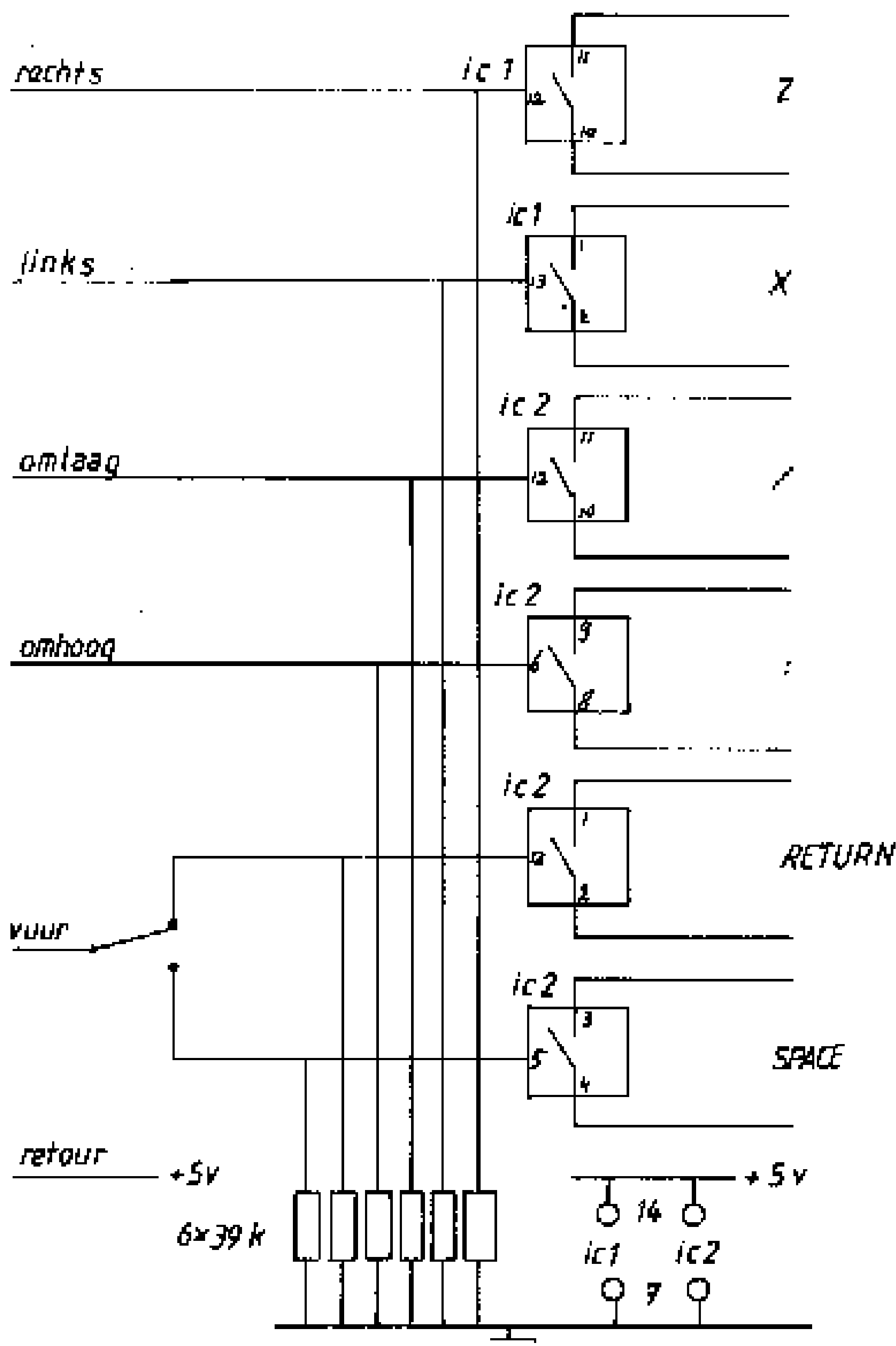
De twee IC's die ik gebruikte zijn van het type 4066. Verder zijn zes weerstanden nodig van 39 kOhm en een omschakelaar. De 4066 bevat vier schakelaars die d.m.v. een stuursignaal aan of uit worden gezet. Dit stuursignaal komt van de joystick en de schakelaar komt parallel aan de toetsen.

De toetskeuze is in principe vrij. Het is aan te bevelen om de "Z", "X", ":" en "/" te kiezen voor links, rechts, omhoog en omlaag. Zo blijft u compatibel met bijna alle bestaande software. Over de keuze van de vuurknop valt te diskuteren: return of spatiebalk. Ik koos de middenweg en nu kan ik met de omschakelaar omschakelen tussen spatie of return.

Het bouwen is vrij eenvoudig. Maak de verbindigen met wat draad, test de schakeling en plak de twee IC's met krachtlijm vast aan de print van het toetsenbord. Zaag vervolgens een gat in de zijkant van de computer voor de D-connector en en boor een gat voor de schakelaar. Klaar is u.

Op deze manier kunt u ten alle tijden spellen spelen die met de toetsen z,x,: en / werken en dat zijn bijna alle spellen!

met de vriendelijke groeten van Roland.


JOYSTICK AAN DE ELECTRON r. leurs
rechts
ic 1
Z
ic1
links
X
ic 2
omlaag
/
ic 2
omhoog
:
ic 2
RETURN
vuur
ic 2
SPACE
retour
+5v
6x39 k
+ 5 v
14
ic1
ic2
7

PC aan de ATOM en RTTY

door Frank Vergoossen

noot van de redactie :
De schrijver van de nu volgende twee artikels was lid van onze club en is een nog steeds liefhebbend atombezitter, getuige de nu volgende 8 pagina's. Het verzoek om zijn kennis over een koppeling PC -> ATOM en RTTY nogmaals in een artikel te willen vastleggen ten behoeve van onze club, werd spontaan ingewilligd. De redactie dankt Frank namens de leden hiervoor.
Heeft U vragen over hetgeen nu wordt beschreven, dan kunt U hem gerust hierover benaderen. De bij deze artikels behorende software staat voor de club vrij ter beschikking.

Op de regioschijven vindt U de hieronder genoemde programma's. De MSDOS-software is uiteraard op een aparte schijf gezet . De programma's die het belangrijkste zijn om naar te kijken, zijn gemerkt met (*). De overige programma's zijn bestemd voor mensen die zelf aan de slag willen met de sourcelistings en dergelijke.

APARTE SCHIJF (MS-DOS FORMAAT):

ASS	BAT	Handige batchfile om snel met de crossassembler te werken tussen PC en Atom. Eventueel voor eigen gebruik iets aanpassen.
BASLOAD	EXE	Converteert een Atombasic programma, geschreven in MS-DOS formaat naar het geheugenformaat van de atom.
BASLOAD	PAS	Turbo-pascal V5.0 sourcecode voor BASLOAD.EXE CSC6502 EXEPublic domain crossassembler voor de 65(C)02
CSC6502	EXE	Public domain crossassembler voor de 65(C)02
PC-ATOM	BAS	Programma voor Atom om vanuit de PC files naar de Atom te zenden.
RTTY	BAS	RTTY Basicgedeelte voor Atom op #6000
RTTY	BIN	RTTY machinecode voor Atom op #7000
RTTY	LST	Output listing van RTTY
RTTY	MXT	RTTY machinecode in Motorola S1-formaat.
RTTY	SRC	RTTY sourcecode.
S1LOAD	EXE	Public Domain programma dat files in Motorola S1-formaat converteert naar binaire code.

REGIO SCHIJF (ATOM-DOS FORMAAT):

PC-ATOM	(*) Assemblerprogramma om files van de PC naar de Atom te sturen.
RTTY	(*) Het RTTY-programma.

PERSONAL COMPUTER AAN ACORN ATOM

Met behulp van twee connectoren en een stuk flatcable, is het mogelijk om bestanden vanuit een personal computer (onder MS-DOS) naar een Acorn Atom te zenden.

Als praktische toepassing kunt u dan bijvoorbeeld denken aan het maken van een machinecodeprogramma voor de Acorn Atom op de PC. Dat kan met behulp van een crossassembler voor de 6502. Na assembleren (op de PC dus) wordt de binaire programmacode naar de Acorn Atom gestuurd. Een ander voordeel is dat dan de geheugenverslindende source-code van het programma niet in het geheugen van de Acorn Atom staat.
Uiteraard kunnen op deze manier ook tekstbestanden of grafische afbeeldingen naar de Acorn Atom gestuurd worden.
Wellicht zijn er nog vele andere toepassingen mogelijk, maar dat laat ik geheel aan uw eigen fantasie over.

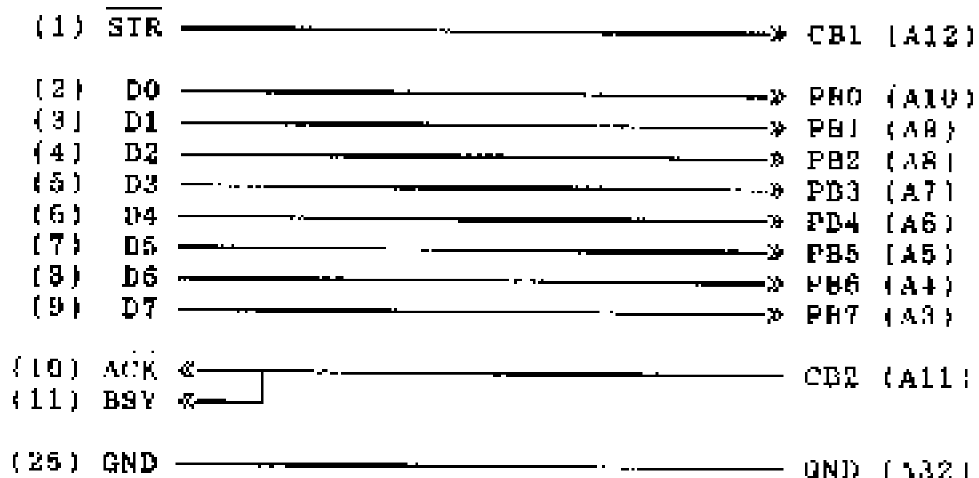
Bestanden van de Acorn Atom naar de PC verzenden is met de hier beschreven aanwijzingen niet mogelijk, maar dat hoeft misschien geen bezwaar te zijn.
Het is wellicht ook mogelijk een andere computer, die over een printerpoort beschikt, te gebruiken als zender. Denk bijvoorbeeld aan een tweede Acorn Atom of een computer van Commodore of Atari. Dit is echter niet door mij getest, en u zult zelf even de juiste aansluitingen op de printerpoort moeten uitpluizen.

Dan volgt nog even in het kort een lijstje met de benodigdheden voor dit moois:

- D-connector, 25 polig male.
- Connector 64 polig, A+B, female recht soldeer.
- Flatcable, minimaal 11 aderig.
- Communicatieprogramma voor Acorn Atom (zie verder).

De bedrading dient volgens het volgende plaatje uitgevoerd te worden. Tussen de haakjes staan de nummers van de pennen op de connectoren. De pijltjes geven alleen de richting van het datatransport aan. Zie eventueel het artikel "Atom bus definitie" in Atom nieuws jaargang 5, nummer 1, pagina 48.

PC (LPT1 of LPT2) ACORN ATOM (VIA 6522)



Op uw Acorn Atom dient het volgende programma ingetypt te worden. De teksten met "REM" en "\" kunt u desgewenst weglaten.

```
 10 REM---------------------
 20 REM  PC AAN ACORN ATOM
 30 REM F. VERGOOSSEN (1989)
 40 REM---------------------
 50 S=#80   ; REM DATA STARTADRES
 60 T=#B000; REM PPI STARTADRES
 70 V=#B800; REM VIA STARTADRES
 80 Y=3; DIM LLY
 90 FOR X=0 TO Y; LLX=#FFFF; NEXT X
100 PRINT $12
110 INPUT "LAADADRES DATA "D
120 INPUT "STARTADRES MACHINECODE "Q
130 PRINT $21; FOR X=1 TO 2; P=Q
140[
150:LL0
160 LDA @D%#100; STA S    \STARTADRES DATA LSB
170 LDA @D/#100; STA S+1  \STARTADRES DATA MSB
180 LDA @#FF; STA V+#D    \INTERRUPTS UIT
190 LDA @0  ; STA V+2     \POORT B IS INVOER
200 LDA @#80; STA V+#C    \HANDSHAKE MODE
210 STA V \DUMMY WRITE RESET CB2
220 JSR #F7D1             \UITVOER TEKST
230]
240 $P="STARTADRES: #"; P=P+LEN P
250[
260 NOP \EINDE TEKST
270 JSR LL3 \UITVOER STARTADRES
280 LDY @0
290:LL1
300 LDA V+#D; AND @#10
310 BEQ LL2 \CB1 NIET ACTIEF
320 STA V \DUMMY WRITE RESET CB2
330 LDA V; STA (S),Y \DATA INVOER EN BEWAAR
340 LDA T+2; EOR @4
350 STA T+2 \INVERTEER SPEAKER
360 INC S   ; BNE LL1
370 INC S+1; BNE LL1
380:LL2
390 BIT T+1; BVS LL1 \TEST CTRL-KEY
400 JSR #F7D1 \UITVOER TEKST
410]
420 $P="EINDADRES : #";P=P+LEN P
430[
440 NOP \EINDE TEKST
450:LL3
460 LDA S+1; JSR #F802 \UITVOER MSB
470 LDA S  ; JSR #F802 \UITVOER LSB
480 JSR #FFED \UITVOER LF+CR
490 RTS \EINDE
500]
510 NEXT X; PRINT $6
520 @=1; PRINT "EINDADRES MACHINECODE: #"&P
530 END
```

Dan volgt nu de procedure om de gegevens van de PC naar de Acorn Atom te senden. Eerst sluit u de connectoren aan op de PC en de Acorn Atom. Dan start u het vorige programma op uw Acorn Atom. Er wordt dan naar het adres gevraagd waar de data wordt opgeslagen, bijvoorbeeld #2800. Dan wordt er gevraagd naar het startadres van de machinecode, bijvoorbeeld #7F00. Daarna kunt u de communicatie starten door te linken naar het startadres van de machinecode, in dit voorbeeld dus: LINK #7F00.
Om dan uiteindelijk de data te verzenden geeft u volgende MS-DOS instructie op de PC:

```
COPY/B FILENAAM LPT1
```

De parameter /B geeft aan dat de gehele file verzonden moet worden. Indien deze /B weggelaten wordt, zou het verzenden stoppen bij een "EOF" code (end of file, asciicode 1A hex.) en dat is meestal niet de bedoeling, vooral niet bij binaire data.
"FILENAAM" is de naam van het bestand op de PC dat naar de Acorn Atom verzonden moet worden. "LPT1" geeft aan dat het bestand naar de printerpoort verzonden moet worden. Het bestand wordt dus als het ware gekopieerd naar de printerpoort. Indien de PC over twee printerpoorten beschikt, dan kunt u "LPT1" vervangen door "LPT2" en de kabel op deze poort aansluiten. De printer kan dan gewoon op poort 1 blijven zitten.

Als de data overgezonden is, wat trouwens in één seconde gebeurd is, drukt u op de CTRL-toets van de Acorn Atom om het programma te onderbreken, en klaar is Kees (of Jan, Piet etc.). De data staat nu in het geheugen van de Acorn Atom en u kunt er mee aan de slag.

Tot slot wens ik u veel succes. Voor vragen kunt u zich altijd tot mij wenden.

Frank Vergoossen, tel. 04754-81972.

KIES UIT (DISK-)DIRECTORY MET JOYSTICK.

G.v.d. Leest.

Geen kant en klare oplossing,maar een eerste aanzet tot een andere werkwijze, pretendeert het programma "BOOTREM" te zijn.
De bedoeling is , met behulp van de joystick een programma te kiezen uit een directory die op het scherm wordt weergegeven.
Met de action-knop kan het geladen en gestart worden.
Op dit moment geldt wel de beperking ,dat het programma met executie-adres #CE86 moet zijn weggeschreven.
Zoals ,gezegd ,het is niet af, en wellicht kan dit uitgangspunt zijn voor een aantal andere programma's.

G.v.d.Leest. redactie.B1.

RTTY TX/RX V2.0 VAN PASELB

INLEIDING

Dit programma is bedoeld om RTTY-signalen (= radio tele type) te kunnen ontvangen en/of uitzenden. Het zal dus in de eerste plaats voor luister- en zendamateurs interessant zijn.

Het programma dat nu beschreven wordt is een verbeterde versie van het RTTY-programma dat beschreven is in "Atom nieuws" nummer 4 van jaargang 6 (1987) pagina 48 t/m 49. Mede dankzij de (opbouwende) kritiek die ik van diverse gebruikers heb mogen ontvangen over de vorige versie (waarvoor mijn dank), is het nu in een ander en nieuwer jasje gestoken. Dat wil niet zeggen dat dit programma niet meer voor verbetering vatbaar is. Voor aanvullende opmerkingen uwerzijds sta ik dan ook graag open.

Voor de geïnteresseerden: De machinecode werd door middel van een crossassembler op een personal computer gemaakt, en via de printerpoort naar de Acorn Atom overgezonden door middel van de VIA-poort.

BENODIGDE CONFIGURATIE

Uw Acorn Atom moet met de volgende hardware uitgebreid zijn:

- RAM van #2800 t/m #9800.
- VIA 6522 inclusief bijbehorende IC's.

Een floating point ROM is niet persé benodigd.

Voor de ontvangst van RTTY-signalen komt er nog een decoder bij die de ontvangen pieptoontjes omzet in logische niveau's voor de Acorn Atom. Aan het einde van dit artikel is literatuur vermeld waarin diverse geschikte schakelingen te vinden zijn.

Aanbevolen wordt een tekstverwerker die tekst vanaf #2800 bewerkt, bijv. ED64, waarmee de ontvangen tekst bewerkt, bewaard en afgedrukt kan worden. Ook kan dan tekst gereed gemaakt worden voor uitzending. In dat laatste geval mogen er geen speciale controle-codes in de tekst voorkomen.

GEHEUGENGEBRUIK

Het geheugen wordt als volgt gebruikt:

#0080 t/m #008F	Zeropage geheugen.
#2800 t/m #5FFF	Tekstbuffer lengte 14 kB, te gebruiken in combinatie met ED64.
#6000 t/m #6BFF	Basicgedeelte (max. 6FFF).
#7000 t/m #76FF	Machinecode (max. 7FFF).
#8200 t/m #97FF	Toetsenbordbuffer lengte 5½ kB.

OPSTARTEN EN BEWAREN VAN HET PROGRAMMA

Het programma kan op de volgende manieren geladen en opgestart worden:

- Vanuit DOS: *RTTY

- Vanuit COS: *RUN "RTTY"

- Vanuit DOS en COS:
  *LOAD "RTTY"
  LINK #7000

- Vanuit DOS en COS:
  *LOAD "RTTY"
  ?18 = #60
  RUN

Het programma kan als volgt bewaard worden:

*SAVE "RTTY" 6000 7700 7000

Na "BREAK" kan het programma opnieuw gestart worden door:

LINK #7000

OPMERKINGEN

Indien het programma alleen maar wordt gebruikt om RTTY te ontvangen hoeft u in het menu alleen maar te letten op: baudrate, polariteit, (un)shift on space en leestekens. De andere keuzemogelijkheden hebben dan geen betekenis. Bij het zenden hebben alle keuzemogelijkheden een functie.

De omschakeling van unshift naar shift en omgekeerd gaat vanzelf na het indrukken van letters of leestekens, dus dat hoeft u niet zoals bij een oude telexmachine zelf te doen.

Tijdens de ontvangst verschijnen er soms geïnverteerde tekens op het scherm, waarvan de betekenis als volgt is:
"b" = bellkarakter, "e" = niet voorkomend leesteken (error).

Tijdens zenden worden niet voorkomende leestekens genegeerd. De LOCK, COPY en de pijltjestoetsen hebben geen functie. REPT kan wel worden gebruikt om een teken herhaald uit te zenden. DELETE verwijdert de laatst ingetypte toets, echter alleen als de toetsenbordbuffer nog niet helemaal is uitgezonden.

De volgende leestekens-sets worden ondersteund:

(Letters)	:	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
INT	:	-	?	:		3			'	8	b	(	)	.	,	9	0	1	4	'	5	7	=	2	/	6	+
WD	:	-	?	:	$	3		&	#	8	b	(	)	.	,	9	0	1	4	'	5	7	;	2	/	6	"
MIL	:	-	?	:	$	3	|	&		8	'	(	)	.	,	9	0	1	4	b	5	7	;	2	/	6	"

Opmerking: de "b" staat voor het bell-karakter.

AANSLUITINGEN

Hier volgen de aanwijzingen hoe de VIA-poort op de apparatuur moet worden aangesloten:

De push-to-talk uitgang zit op poort PB6, en hiervoor geldt dat TX = "0" en RX = "1". Deze uitgang wordt gebruikt om uw zendontvanger om te schakelen op zenden. Indien u per ongeluk op "BREAK" drukt, dan zal de zendontvanger niet gaan zenden omdat PB6 dan als ingang wordt geschakeld en daarom "1" wordt. Bij de vorige programma-versie was dit andersom.

De ontvangst-interface voor RTTY wordt aansluiten op poort PC3 van de PPI 8255. Deze zit op pen 7 van de cassettepoort.
Het is ook mogelijk om hiervoor poort PB0 van de VIA 6522 te gebruiken. Als dat het geval is, dan zult u in het basicgedeelte op #6000 de waarde van een variabele moeten veranderen (commentaar is in het programma bijgevoegd).

De AFSK-uitgang (audio frequentie shift keying) zit op poort PB7 van de VIA 6522. Als u wilt dat de signalen verschijnen op de uitgang van de cassettepoort, dan kunt u deze aansluiting via een weerstand van 4K7 aan deze uitgang hangen, net als bij het populaire programma "HJCOS". Dat is ook handig als u de signalen op de cassetterecorder wilt opnemen.

Als de signalen de zendontvanger in gaan zal het TTL-niveau op deze uitgang eerst verzwakt moeten worden. Dat kan met een spanningdeler bestaande uit twee weerstanden. Door middel van een condensator zal de blokgolf wat sinusvormiger gemaakt moeten worden om de uitgezonden bandbreedte te begrenzen.

BEDIENINGSFUNCTIES

Hier volgt in het kort het gebruik van het programma.
De meeste functies zullen wel duidelijk worden nadat u ze een keer heeft gebruikt. Een kleine letter betekent dat zowel de SHIFT als de bedoelde letter tegelijk ingedrukt moeten worden.

In menu

t of T: ga direct naar zendroutine (transmit).
r of R: ga direct naar ontvangstroutine (receive).
1...9 : veranderen ingestelde waarden.

Tijdens zenden én ontvangen

ESC: ga terug naar het menu (escape).
i : bewaar vanaf nu alle tekst in de buffer vanaf #2800 (input).
p : stop de tekstbuffer input of output (stop).

Tijdens ontvangen

t: ga naar zendroutine (transmit).

Tijdens zenden

r: ga naar ontvangstroutine (receive).
b: bell-karakter (bell).
e: ga naar ontvangstroutine na uitzenden tekst (end).
o: uitzenden tekstbuffer vanaf #2800 (output).

Polariteit MLSH heeft de volgende betekenis:
bij TX: mark lage toon, space hoge toon.
bij RX: mark = "0", space = "1".

Polariteit MHSL heeft de volgende betekenis:
bij TX: mark hoge toon, space lage toon.
bij RX: mark = "1", space = "0".

Blank teken aan: tegenstation is beter gesynchroniseerd en telexmachine van de ontvanger blijft ingeschakeld tijdens lange type-pauzes.

Auto CR/LF aan: bij zenden wordt na 63 tekens automatisch CR/LF gegeven in plaats van de dan ingetypte spatie, doch uiterlijk na 69 tekens wordt automatisch een CR/LF tussengelast. Dit maakt het mogelijk gebruikers van een telex-machine het bericht mee te laten volgen, zonder dat de machine aan het einde van de regel blijft "hangen".

Unshift on space (UOS) geeft een unshift nadat een spatie wordt ontvangen. Dit verkleint de kans op fouten tijdens de ontvangst van veel tekst en weinig leestekens.
Shift on space (SOS) geeft een shift na het ontvangen van spatie. Dit verkleint de kans op fouten tijdens de ontvangst van veel leestekens of getallen en weinig tekst, zoals bij weercodes.

In het programma is een snelle VDU-routine ingebouwd, nog sneller dan bij de vorige versie. Dit heeft tot gevolg dat de tekst gewoon omhoog schuift als het scherm vol is, en niet opnieuw begint met een leeg scherm zoals bij de vorige versie.

Tevens is een nieuwe printerroutine ingebouwd. Om de tekst te printen hoeft u alleen de printer aan te zetten en deze op "on line" te zetten. Het programma merkt zelf wanneer de printer aan staat, en zorgt dan dat de uitvoer ook naar de printer gestuurd wordt.

Tijdens het zenden verandert de cursor in een "t" (transceive), tijdens ontvangen in een "r" (receive).

GEBRUIK BASIC-VARIABELEN

A Auto CR/LF, 0=UIT/1=AAN.
B Tellerstand (16 bit) a.h.v. baudrate en clockfreq.
C Clockfrequentie v.d. Atom, normaal 1000000 Hz.
D Dummy variabele voor diversen.
E Lengte stopbit, 0=1 BIT/1=1,5 BIT/2=2 BIT.
F Leestekens, 0=INT/1=WU/2=MIL.
G Blank teken, 0=OFF/1=ON.
K Poortkeuze bij ontvangen 0=VIA PB0/1=PPI PC3
L Laagste frequentie, 1275/2125 Hz.
M Tellerstand (16 bit) a.h.v. laagste frequentie, frequentieshift en clockfrequentie. M is normaal MARK.
N Tellerstand (16 bit) a.h.v. laagste frequentie en clockfrequentie. N is normaal SPACE.
P Polariteit, 0=MLSH/1=MHSL (M en N worden verwisseld).
Q Startadres machinecode [#7000].
R Baudrate, 45/50/57/75/100/110 Baud.
S Frequentieshift, 85/170/300/425/850/1000 Hz.
T Toetsleesroutine.
U (Un)shift on space, 0=OFF/1=UOS/2=SOS.
W Waarde ingelezen toets in variabele T (0 t/m 9).

Niet gebruikte variabelen: H,I,J,O,V,X,Y,Z.

GEBRUIK ZEROPAGE

Startadres zeropage #80.

Bytes Betekenis

1 Baudotcode.
1 Tweede Baudotcode na (un)shift of CR (dan LF)
1 Asciiwaarde af te drukken karakter.
1 0=letters, 1=figures, 2=(nog) geen van beide.
1 Asciiwaarde laatste ingedrukte toets, 0=geen.
1 Aantal tekens na laatste CR/LF.
1 Eindwaarde voor teller bij zenden van figures.
2 Teller bij ontvangen.
2 Leespositie toetsbuffer.
2 Schrijfpositie toetsbuffer.
2 Schrijf/leespositie tekstbuffer.
1 Tekstbuffer; 0=nonactief, 1=input, 2=output.

STARTADRESSEN MACHINECODE

Startadres programma #7000.

Offset Betekenis

#00 Start basicprogramma op #6000.
#0F Init VDU & printer.
#12 Init VIA & zeropage-RAM.
#15 Herstart tonen.
#18 Ontvangstroutine.
#1B Zendroutine.
#1E Toetsleesroutine naar variabele T.

LITERATUUR

Voor schema's van ontvangst-decoders voor RTTY en artikelen over RTTY kunt u de volgende bladen en/of boeken raadplegen:

Elektuur juli/augustus 1982 blz. 8-14.
Elektuur juni 1983 blz. 6-37, 6-68, 6-72.
Elektuur infokaart 84, normen 12 "telegrafie-kodes".
Elektuur infokaart 87, normen 13 "Morse-tekens".
Elex september 1986 (?), RTTY-konverter blz. 9-14.
Acorn Nieuws samenvatting 1982 blz. 96.
Atom Nieuws jaargang 8, nummer 4, pagina 46 t/m 49, RTTY TX/RX.
Vademecum VERON 7* druk blz. 279.

SLOTWOORD

Het programma zal verspreid worden via het diskette- en bandjesarchief van de Atomclub. Indien u geïnteresseerd bent in de source-listing en aanvullende informatie, dan kunt u contact opnemen met Jan Bronzwaer of ondergetekende. Ook voor problemen met betrekking tot het gebruik van het programma kunt u zich tot mij wenden.

Tot slot wens ik u veel succes met het gebruik van het programma.

Frank Vergoossen, roepletters PA3ELB, telefoon 04764-81872.

S P S Sanders Print Service

Cassette interfaceprint	5.00
MDCR interface print	5.00
Battery backup printje	3.00
8K hoge geheugenprintje	5.00
#E000 naar #1000 voor oude schakelkaart	3.00
Omschakelprintje voor de 80K kaart	3.00
Bootstrapprintje de Moor	6.00
Voedingsprint MDCR 12V	5.00
Acoustische verbindingstester	5.50
Combikaart	30.00
Viaprint Z80 (Atombus)	15.00
8e printerbit	7.00

Al deze printen zijn ook gebouwd en getest te bestellen tegen kostprijs onderdelen en een symbolisch bedrag. Even bellen graag. (Tel. 04750-30401)

Bestellen: op de clubavond te Sittard of door overmaking van het bedrag (Fl.- per print verpakkings- en verzendkosten) onder vermelding van de print(en) op gironr: 794739 tnv: E. Sanders. Rozenlaan 13 te Herten.

PROBLEMEN MET "AUTO-LINEFEED"

H.Bastings.

Echte fanaten,die van meerdere computers gebruik maken,die weten dat vele computers de printer standaard GEEN AUTO-LINEFEED laten verzorgen.
Gebruik je dus meerdere machines op een printer ,dan moet je meestal onder een of ander klepje van de printer ,of op een andere onbereikbare plaats ,een schakelaar of dipswitch om zetten.
De ATOM geeft,zoals velen zullen hebben gemerkt,standaard een auto-linefeed,die echter gemakkelijk software-matig kan worden uitgeschakeld , door in #FE de waarde 0 te schrijven.
Een nadeel is dat je steeds,na het opstarten of na een BREAK, deze waarde opnieuw in #FE moet schrijven.
Wil je dit voortdurend goed zetten voorkomen,en is er verder voor het ATOM-gebruik geen behoefte aan de standaarduitvoering,dan kan dit definitief veranderd worden door in de RESET-ROUTINE in adres #FF5C in plaats van #0A,#00 te schrijven.

Veel succes.
H.Bastings ,Regio Limburg.

A T O M M A R K T

Te koop AANGEBODEN:

Acorn Atom Computer.
Te bevragen bij dhr. Soepboer.
Tel.: 03200 - 30556

Star printer type DP510
Prijs 250,--
Te bevragen bij dhr. G. Hermans
Puntelstraat 24, Simpelveld
Tel. 045 - 444590

Te koop GEVRAAGD:

Een monochrome monitor voor de Atom.
Een diskdrive voor de Electron (of het schema ervan)

Alle reacties graag naar: Roland Leure
Nattenhoverkoestraat 5
6129 LH Nattenhoven
Tel. 04490 - 36454

```
******************************
*      W R I T E - III       *
******************************
```

door Jeroen van der Veen

Write-3 is, niet zoals u misschien denkt, een nieuwe tekstverwerker, het is een tekenprogramma, dat gespecialiseerd is in het zetten van letters op het grafische scherm, om hiermee bijvoorbeeld tekeningen van teksten te voorzien. Het programma is eigenlijk gemaakt als aanvulling op het tekenprogramma Macintosh, dat ik ,aangepast voor GDOS, regelmatig gebruik. Dit is een mooi, snel en vrij veelzijdig programma, ik miste echter de mogelijkheid om letters te zetten en een screendump te maken. Het programma Macorn kan dit wel, maar de snelheid van dat programma speelde mij al snel parten, waarna ik het aan de kant heb gezet. Oorspronkelijk was Write een simpel programmaatje om letters, gemaakt door de spriteeditor, op het scherm te zetten. Het programma groeide meer en meer uit tot wat het nu is. Write-III is volledig menugestuurd, en werkt met vensters(!!!). Bovenaan het scherm staat een menubalk (a la ST) waar je met je joystick je handje naar toe kunt bewegen, en je keuze te maken door op de juiste keuze te gaan staan, en vervolgens de vuurknop in te drukken. Het scherm wordt dan geunflowd, en er verschijnt een venster op het scherm. Door met de joystick het goede item uit te kiezen, en daarna op de vuurknop te drukken, wordt dit onderdeel geactiveerd. Voor de meest gebruikte optie hoef je echter niet eerst een venster op te roepen, of zelfs naar de menubalk te gaan. Het zetten van tekst op de positie van het handje (de pointer) gaat zeer simpel, je kiest je plaats en klikt de vuurknop in, het scherm gaat over in de tekst-mode, je typt de gewenste tekst in, drukt op return, en voila, daar staat je tekst. De menubalk ziet er als volgt uit :

```
-------------------------------------------------
| NEW  | DRAW | PRN. | DISC | FONT | MISC |
-------------------------------------------------
```

New : Er verschijnt een venster met de volgende keuzes :
- -Restart : Scherm wordt schoongemaakt, alles wordt weer gereset.
- -Quit : Verlaat programma.
- -Cancel : Verlaat venster zonder iets te doen, verschijnt in ieder venster, dus verder wordt er geen melding van gemaakt.

Draw : Er kan gekozen worden uit:
- -Erase : Pointer verschijnt omgekeerd op het scherm, als je op de vuurknop drukt wordt er een stukje gewist, bewegen met joystick
- -Scroll : Scherm scrollt met beweging van joystick mee, afbreken met vuurknop

Prn. : Printerdump menu, de keuzes zijn:
-Big Pdump : Grote formaat printerdump, TX/RXbox nodig !
-Small Pdump : Klein formaat printerdump, TX/RXbox nodig

Disc : Disc-acces menu, Gdos nodig, Ados versie komt !
-Save screen : Schrijft huidig plaatje weg naar geselecteerde drive
-Load screen : Laad plaatje in het videogeheugen in.
-Catalogue : Vraag drivenummer en geeft catalogus
-Qualifier : Toont huidig qualifier en vraagt om de nieuwe
-Cancel : Verlaat venster zonder wat te doen

Font : Diverse mogelijkheden voor andere letters
-Underline : Underlined de eerstvolgende tekst die gezet wordt
-Char.Space : Tekstmode, vraagt om characterspace (default 6)
-Load Font : Laad fontfile in op #5400, zie verdere uitleg.

Misc : Miscellaneous, diverse grappen
-Invert : Inverteert het scherm.
-Mirror : Het scherm komt op de kop (!!) te staan
-Edge-detect : Voert een egde-detect uit (Van G.Hildebrand)
-Border : Tekent een lijstje om de tekening.

Verdere uitleg Fontfile

De fontfile is een basicprogramma, dat voornamelijk gevuld is met het gags-statement CREATE, hierin bevindt zich de informatie voor de letters. Er kunnen echter maar 54 letters in een karakterset staan, omdat gags maar 55 sprites tegelijk aankan, en er is een sprite nodig voor de pointer (het handje)
Bij WRITE-III zijn 3 fontfiles geleverd, namelijk :
NORMfnt - Het standaard karakterset
ITALfnt - Een cursief karakterset
SHADfnt - Een "shadow" karakterset

Extra fonts zijn gemakkelijk te maken met behulp van de Sprite-Editor van R.Boers. T.z.t. zal ik nog wat meer fonts maken, en die zullen wel weer verspreid worden. Zelf fontfiles maken is makkelijk, als u dit aanhoudt:

```
10 REM naam fontfile
20 BASE #98 (of een andere page)
30 CREATE A,#65,#20,0,3,5,7,2,6,7,3,5,6,#20,4,#56,#13\A:CB"A"
enz.
500 ?18=#29;RUN
```

Let er op dat de laatste regel in ieder geval in de fontfile staat, anders wordt er niet teruggekeerd naar het hoofdprogramma. Bedenk wel als u een letter wil zetten die niet gedefinieerd is, het programma een error geeft !
De volgende karakters zijn in mijn fontfiles ondersteund :
ABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890,.+?/:!"#$%'()=-*
N.b. Een fontfile begint altijd op #5400

Benodigd voor Write-III :

Write-III stelt (helaas) hoge eisen aan uw atom, nodig is:
a - een volgens clubnorm aangesloten joystick
b - geheugen van #2900-#9FFF
c - Een discdrive, liefst Gdos, Ados versie komt nog.
d - Gags en F-charms, dus ook goede schakelsoft.
e - Voor de screendumps heeft u de TXbox nodig, een door onszelf
    aangepaste versie voor de G.E.txp-1000 van de RXbox.
N.B. Het programma past zichzelf houdt er rekening mee of er een 80-koloms vdu is of er een standaard vdu32 is. Zie regel 30 van het hoofdprogramma, hier staat een vlag, die gezet kan worden op 0 (vdu80) of 1 (vdu32).

Op de regioschijf vindt U :

Het hoofdprogramma	:	WRITE-3
Een assemblergedeelte	:	WRITmcb
het afgebeelde plaatje	:	WRT3pic
de vier karaktersets	:	NORMfnt - STRTfnt - ITALfnt - SHADfnt

WRITE-III DEMOSTRATIE

ABCDEFGHIJKLMNOPQRSTUVWXYZ

ABCDEFGHIJKLMNOPQRSTUVWXYZ

ABCDEFGHIJKLMNOPQRSTUVWXYZ

VOOR EDGE-DETECT

ABCDEFGHIJKLMNOPQRSTUVWXYZ

ABCDEFGHIJKLMNOPQRSTUVWXYZ

ABCDEFGHIJKLMNOPQRSTUVWXYZ

K A R A K T E R -SPACE

DIVERSE AANVULLINGEN / VERBETERINGEN.

B.Meintema.

1. Infomaster met labelfunctie en pijlbesturing

Naar aanleiding van het programma "Infomaster-80 ,heb ik ook dit programma nog iets aangepast. Ik heb er nog een mogelijkheid ingebracht om labels te printen.
Bij het instellen van de printer wordt nu ook gevraagd "LABEL ?". Indien U hier "J" antwoord, wordt gevraagd "AANTAL ?".
Hier kunt U het aantal linefeeds tussen de labels instellen.
Bij de instelling "items printen ",wordt ook weer gevraagd "LF?"
Hier dient U ook met "J" te antwoorden.

Voor de aardigheid heb ik ook nog een versie met pijlbesturing gemaakt. Dit werkt ook wel leuk. U moet het maar eens proberen.
INF801a is de versie met labelmogelijkheid.
INF801m is de versie met label- en pijlbesturing.
Voor beide versies geldt dat men geheugen van #600 tot #A00 moet hebben.
Bij versie INF801m is de file MENUas2 nodig;
Bij versie INF801a is de file IMCODE1 nodig.

2.Verbetering van file-conversie van ATOM- naar GDOS.

Bij het gebruik van het conversieprogramma AD>GD (AN 5-2) beviel het mij niet ,om steeds als ik een schijf had gecopieerd ,de start- en eindadressen te moeten aanpassen.
Na bestudering van het programma DCOPY ,ben ik er in geslaagd om het programma zo te veranderen,dat de adressen nu meteen goed staan.
Bij het vraagteken achter de file-naam moet men "N" en return geven ,als de file niet gecopieerd moet worden.
Copieren met dezelfde naam door een return te geven,danwel de nieuwe naam + return.
Ook bestaat de mogelijkheid de gehele schijf direct te copieren.
Hopelijk hebben een hoop mensen hier plezier van.
De file-naam is nu "AD>GD,2"

3. Het Volkslied.

Dan nog het programma "WILHELM".Ik heb dit aangehoord met tranen in de ogen.Het getuigd m.i. van weinig kennis van het Nederlandse Volkslied. Het is namelijk niet compleet.De "Vaderlandsliefde" gebood mij dit lied te repapreren;hierbij de goede melodie.

4. Nieuws over VDU-3.3.

Het VDU-3.3 programma van P.Wokke heb ik gebrobeerd, en bij de eerste opstart werkte het voortreffelijk.
Maar ..... na een break , en hernieuwde opstart wilde het scherm niet meer scrollen !.
Na wat proberen is het mij gelukt ook dit in orde te krijgen.
Door bij initialisering te zorgen dat het adres #BE25 nul is , werkt het verder goed en zijn er leuke statements voorhanden.
HELAAS !!!! : de meeste programma's geschreven voor VDU 2.7 of VDU 2.8 werkten niet goed meer en moeten worden aangepast.JAMMER!
(Zie ook AN 8-1 blz.23 van J.Jobse !!!).

5. Een sfeervol programma .

Bij het doorzoeken van mijn CATALOG,vond ik nog een gerafisch programma ,dat bij mijn weten nooit is gepubliceerd ;daarom doe ik het hierbij , Het is geschreven door GUIDO de ROOS.
Ik kan alleen maar zeggen,dat het sfeerscheppend is.
Het past heel goed bij een glaasje wijn en een mooie vrouw.
Draai het programma eens,maar laat het wel een poosje draaien om het effect te bekijken. Het programma heet "KAARS".

```
******************
*   H E L P !   *
******************
```

Kan iemand mij enige informatie geven over een of meer van onderstaande ic's ? Reacties graag naar :

Roland Leurs
Nattenhoverkoestraat 5
6129 LH Nattenhoven
tel 04490-36456
of naar de redactie van Atom Nieuws.

P8085	P8155	P8212	P8251a
P8253	P8257	P8316	P8355

Deze ic's komen uit een terminal, ADDS REGENT 20.

```
*******************************************
*               M O D E M                 *
*******************************************
```

door Roger Boussen

Het programma MODEM vormt samen met LOGIN, INIT-LOGIN en ASS-LOGIN een hulpmiddel indien je het ingeven van de codes bij de verschillende databanken ook zo erg vindt. Vooral de FIDO-achtigen kosten een hoop werk en tijd.
Met het prachtige VDT-6.0 (zie AN.7-4 , red.) is hieraan een einde gekomen door de mogelijkheid van auto-transmit <CTRL><5> en een externe functie aanroepen <CTRL><0>.
De codes worden per stuk ingegeven en opgeslagen door het programma INIT-LOGIN. Dit programma bevat een beknopte uitleg waarin staat dat je moet eindigen met <RETURN><RETURN>. Dit is gedaan om het mogelijk te maken een <RETURN> in je code mee te geven.
Na het aanmaken van de verschillende codes moet je in het programma LOGIN de namen van de databanken nog achter de juiste nummers zetten. LET OP dat je de VIDITEL en ASCII namen in het juiste gedeelte zet. Eerst de VIDITEL, zie :viditel, dan de ASCII, zie :ascii.
Nu run je LOGIN, dit programma verwacht SALFAA-1.5.
LOGIN produceert een machinetaal-routine, en list de uitdraai op het scherm.
Als je de databank-namen op de puntjes (en niet verder) hebt ingevuld, geef dan na het runnen *SAVE ASS-LOGIN 7800 7FD8.
Anders even na de laatst geliste mnemonic kijken voor het eindadres.

Indien je nu gaat "modemen" tik je *MODEM, let op: de disc moet in drive 0, waarna gevraagd wordt of je een ASCII of een VIDITEL bank wilt. Na het antwoord wordt de VDT-6.0 soft hierop ingesteld en krijg je het info-scherm van VDT-6.0.
Met <CTRL><0> kom je nu aan de lijst met databanken waarvan je nu de code kunt lezen. Hierna kun je deze met <CTRL><5> verzenden.

Het klinkt allemaal moeilijker dan het in de praktijk is.
Voor eventuele vragen of verbeteringen : bel gerust.

Roger Boussen,

```
******************************************************
*      LINEAIRE NETWERK ANALYSE  LINEAN-V2.3         *
******************************************************
```

door Bert van der Veen

Toen ik, intussen al weer geruime tijd geleden, enkele analoge schakelingen wilde doorrekenen, heb ik daarvoor het programma LINEAN van Kien Stoop en Maarten van Alphen gebruikt. Dat is een programma voor kleinsignaal analyse voor electronische schakelingen. Een beschrijving van dit programma is te vinden in Acorn Nieuws 3-7 (1984).Dit programma stamt uit de tijd dat de geheugenomvang in het algemeen zeer beperkt was en is om die reden zo compact mogelijk geschreven.
De uitvoer van de berekeningen kwam alleen in tabelvorm op het scherm beschikbaar. Ik wilde de resultaten achter vastleggen, maar om die nu van het scherm over te schrijven...?

Zo ben ik begonnen het oorspronkelijke LINEAN uit te breiden met een uitvoermogelijkheid naar de printer. Daarna leek het toch ook mogelijk te moeten zijn de resultaten grafisch weer te geven, m.b.v. het intussen verschenen GAGS van Gerrit Hillebrand. Gelukkig had Kien Stoop nog een becommentarieerde versie van LINEAN beschikbaar, zodat de draad wat beter te pakken was te krijgen, maar het oorspronkelijk programma heb ik nagenoeg ongewijzigd gelaten. Alle uitbreidingen zijn om het echte LINEAN heengebouwd, zodat niet bepaald kan worden gesproken van een gestructureerd programma. Maar desondanks is het toch een nuttig geheel gebleken, waarmee het mogelijk is analoge schakelingen uitgebreid door te rekenen, en de uitkomsten op papier vast te leggen.

Veranderingen t.o.v. het oorspronkelijke LINEAN;

1. De circuitbeschrijving wordt niet meer achter LINEAN opgenomen, maar als een apart programma opgeslagen vanaf #9800.
2. In de AUTO mode wordt vooraf om het gewenste aantal decaden gevraagd, zodat niet telkens na 15 regels wordt gewacht op een toetsaanslag. Nadat het gekozen aantal decaden is afgewerkt kan evt. nog verder worden doorgegaan.
3. Het toegestane frekwentiegebied is uitgebreid van 1 mHz tot 1000 GHz.
4. Nadat de berekeningen zijn uitgevoerd, kan de output naar de printer worden gestuurd, en wordt gevraagd of er grafieken moeten worden getekend. Deze verschijnen eerst op het beeldscherm, waarna evt. een dump op de printer kan worden gemaakt.
5. Na voltooien en evt. dumpen van de grafieken wordt nogmaals gevraagd of de tabel alsnog naar het scherm of de printer moet, waarna indien gewenst ook de circuitbeschrijving kan worden geprint.

6. Na beeindigen van het programma wordt de Basic pointer naar #9800 verplaatst, zodat direct een listing van het circuit kan worden verkregen en evt. wijzigingen in het circuit kunnen worden aangebracht.
7. Nog diverse wijzigingen die ik me niet zo gauw herinner, maar zichzelf wijzen.

Enkele opmerkingen over eigenaardigheden:

Het programma heeft PCHARME en GAGS nodig en daarmee ook schakelsoft (Brenquart doet het bij mij goed). Er treedt echter af en toe een fout op, op het moment dat er floating point berekeningen worden uitgevoerd terwijl GAGS op het A-blok staat. Ik heb lang naar de oorzaak gezocht, maar nooit de echte reden kunnen vinden. Het vermoeden leeft dat de problemen hebben te maken met het gebruik van zero-page adressen door GAGS, welke als floating point werkgebied staan genoteerd (#60 - #7F). Omdat het me te ver ging GAGS hiertoe te wijzigen, wordt op die plaatsen waar floating point berekenigen worden uitgevoerd zoveel mogelijk PCHARME geforceerd door enkele loze statements.
Verder maak ik gebruik van de dumproutine uit de TXBOX, een voor de General Electric TXP-1000 aangepaste versie van de RXBOX. Deze kan echter eenvoudig worden vervangen door een andere dumproutine.

Daar niet iedereen zal beschikken over de gebruiksaanwijzing uit AN 3-7 en er bovendien enkele wijzigingen zijn, geef ik de aangepaste versie er nog maar eens bij. Tenslotte enkele voorbeelden van schakelingen met de bijbehorende uitvoer.

1. BESCHRIJVING:

LINEAN is een LIneair NEtwerk ANanlyse programma voor klein signaal analyse van electronische analoge schakelingen. Het programma berekent de frekwentie afhankelijkheid van de spanningsoverdracht en knooppuntimpedanties.
De werking berust op het vertalen van de circuitdata in een complexe admittantiematrix, welke met Gauss eliminatie wordt opgelost.

2. SPECIFICATIE:

```
Frekwentieschaal: LIN  lineair, start, stop, stapgrootte
                  LOG  logaritmisch, start, stop, aantal
                       stappen
                  AUTO logaritmisch, op te geven aantal
                       decaden, in vaste reeks 1, 1.5, 2,
                       3, 4, 6, 8, 10;  startdecade opgeven
                       (0=1E0; 3=1E3; -3=1E-3)
```

Output: AU spanningsoverdracht; modulus, argument
ZD knooppuntimpedantie; modulus, argument

Elementen: R weerstand
C capaciteit
L zelfinductie
T transistor
A operationele versterker
J spanninggestuurde stroombron

Uitvoer: Naar beeldscherm en optioneel naar printer, in tabel- en/of grafische vorm

Geheugen: Hoofdprogramma #2900-#5500
Circuitbeschrijving vanaf #9800
Werkgeheugen vanaf #5600 tot zover mogelijk

3. ELEMENT SYNTAX:

```
Nx = knooppunt x; Nr = volgnummer
Weerstand    : R<Nr>,<Na>,<Nb>,<waarde in Ohm>
Zelfinductie: L<Nr>,<Na>,<Nb>,<waarde in Henry>
Capaciteit   : C<Nr>,<Na>,<Nb>,<waarde in Farad>
Transistor   : T<Nr>,<Ncoll>,<Nbasis>,<Nemitt>,
               <steilheid in A/V>, <stroomversterking>,
               <afsnijfrekwentie in Hz>
Op.Amp       : A<Nr>,<N+input>,<N-input>,<Noutput>,
               <DC versterking>, <3dB punt in Hz>,
               <uitgangsweerstand in Ohm>.
Stroombron   : J<Nr>,<N+spann.input>,<N-spann.input>,
               <Nstroomsink out>,<Nstroomsource out>,
               <overdracht in A/V>
```

N.B. Voor een transistor geldt voor de steilheid Gm = 40*Ic, bij kamertemperatuur; Ic is de collector(gelijk)stroom. De stroomversterking is de alom bekende Beta. Bij de afsnijfrekwentie is de versterking tot 1 keer gedaald. Bij de Op.Amp is de DC versterking de Open Loop Gain, bij het 3 dB punt is deze 3 dB gedaald.

4. CIRCUIT BESCHRIJVING:

Circuitbeschrijving begint op #9800	
Eerste regel bestaat uit:	10 REM circuitnaam
Tweede regel bevat label d:	20dR1,2,3,470
Volgende regels spatie na regelnr:	30 R2,4,5,4.7E3
Laatste regel afsluiten met END of E.:	90 END
Slechts een element per regel	
Geen knooppunt nummers overslaan	
Element syntax nauwkeurig aanhouden	

5. WERKWIJZE:

Teken het schema.
Nummer de knooppunten met aaneensluitende getallen. Het referentie knooppunt moet het nummer 0 hebben, dit is meestal de voedingsnul. Alle spanningen worden ten opzichte van dit knooppunt uitgerekend. Denk er aan dat ook alle voedingsknooppunten aan 0 moeten liggen, omdat het een AC analyse is. Sla geen knooppuntnummers over. Voer vervolgens de circuitbeschrijving in volgens pnt. 4, dus op #9800.
Zet de Basictextpointer terug op #2900 (?18=#29) en run het programma.
Beantwoord de gestelde vragen, waarna de berekeningen worden uitgevoerd.
Na het rekenwerk kan de uitkomst op diverse manieren worden bekeken en evt. worden vastgelegd. Na afronding hiervan kan de circuitbeschrijving weer worden gelist en evt. voor een volgende rekenronde woden gewijzigd.

6. OPMERKINGEN:

1. Denk er aan dat de benodigde geheugenruimte zeer snel toe neemt met het aantal knooppunten en dat het rekenwerk dan ook steeds langer gaat duren.
2. Komt een knooppuntnummer te vervallen (b.v. door het verwijderen van een R-C serieschakeling), hang dan hieraan een dummy element (b.v. een weerstand van 1E12 Ohm).
3. De snelheid vn het programma is het hoogst door de output het hoogste en de input het op een na hoogste knooppuntnummer te geven. Dit hangt samen met de matrix oplosmethode. Het is echter niet noodzakelijk.
4. Voor het programma bestaat er geen verschil tussen een PNP en NPN transistor, omdat het slechts een AC analyse betreft.
5. M.b.v. de spanniggestuurde stroombron kan men eenvoudig weer nieuwe elementen maken naar eigen ontwerp, door deze weer als deelschakeling uit te tekenen (b.v. een FET).
6. De volgnummers van de componenten hebben voor het rekenwerk geen betekenis, maar dienen alleen voor het gemak van de gebruiker. Bestaande nummering van een schema kan simpelweg worden aangehouden.
7. Denk bij het berekenen van ingangs- en uitgangsimpedanties er aan het andere knooppunt met de juiste weerstand af te sluiten (geen 0 Ohm!).

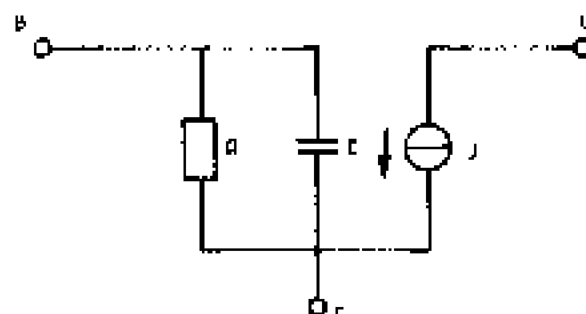
Software :

Op de regioschijf vindt U :

LINEAN2	Het LINEANprogramma , versie 2.3
BANDDRL	Demoschakeling banddoorlaatfilter (laden op #9800)
ATOMAMP	Idem Atom Cassette input-versterker
T-RESON	Idem Dubbel-T resonantiefilter

7. MODELLEN:

7.1. Transistor:



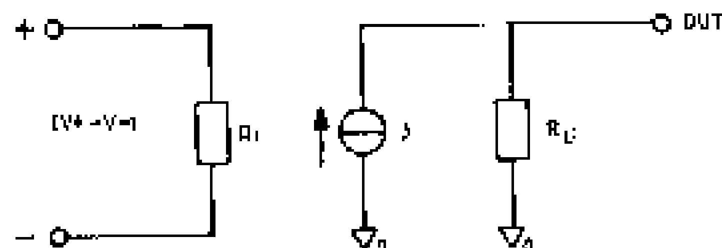
Invoer:

Gm steilheid (40*Ic)
Beta stroomversterking
Ft afsnijfrequentie

Model:

R = Beta/Gm
C = 2*pi*Ft/Gm
J = Gm*(Vb-Ve)

7.2. Operationele versterker:



Invoer:

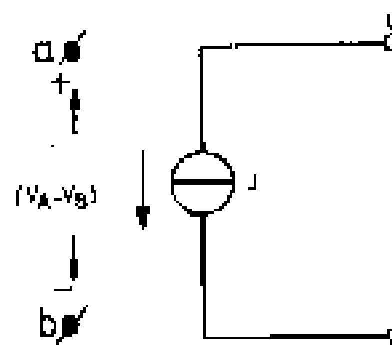
A0 DC versterking
Fc Kantelfrequentie (-3dB punt)
Ru Uitgangsweerstand

Model:

Ri=1E6 Ohm
Ru=Ru
J=A*(V+ - V-)/Ru
waarin A een complexe versterking is:
A=re.A + j*im.A
re.A=A0/(1+(2*pi*F/2*pi*Ft)^2)

```
                A0*(2*pi*F/2*pi*Ft)
im.A  =   -   -----------------------
              (1+(2*pi*F/2*pi*Ft)^2)
```

7.3. Spanninggestuurde stroombron:



Invoer:

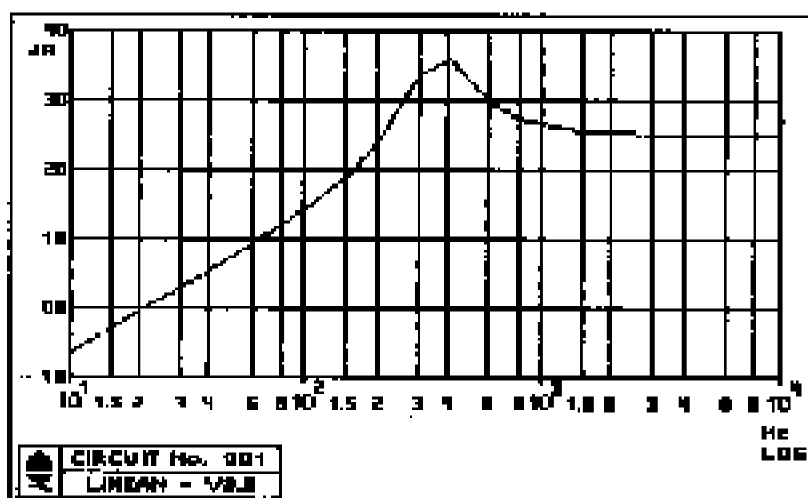
Gm Overdrachtsfactor

Model:

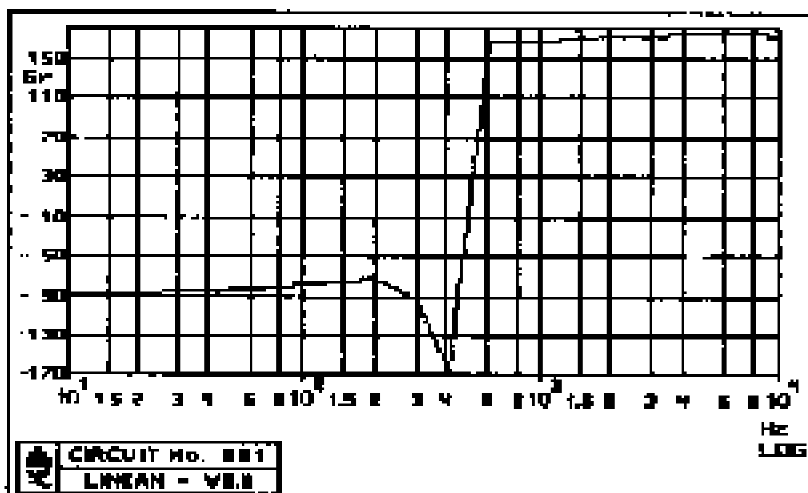
J=Gm(Va - Vb)

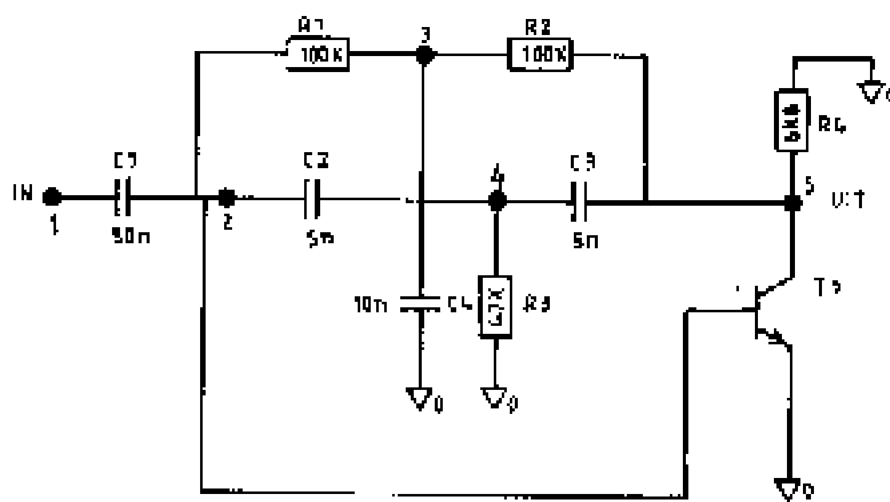
```
CIRCUITNAAM: T RESON
CIRCUIT NO. = 1

10.00  HZ     -8.4 DB     -89 GR
15.00  HZ     -7.8 DB     -88 GR
20.00  HZ      -.3 DB     -88 GR
30.00  HZ      3.3 DB     -86 GR
40.00  HZ      5.8 DB     -85 GR
60.00  HZ      9.6 DB     -83 GR
80.00  HZ     12.4 DB     -81 GR
100.0  HZ     14.7 DB     -79 GR
150.0  HZ     19.8 DB     -75 GR
200.0  HZ     24.1 DB     -74 GR
300.0  HZ     34.0 DB     -93 GR
400.0  HZ     36.3 DB     167 GR
600.0  HZ     29.7 DB     167 GR
800.0  HZ     27.6 DB     167 GR
1.000 KHZ     28.7 DB     169 GR
1.500 KHZ     25.8 DB     172 GR
2.000 KHZ     25.6 DB     173 GR
3.000 KHZ     25.4 DB     175 GR
4.000 KHZ     25.3 DB     175 GR
6.000 KHZ     25.2 DB     175 GR
8.000 KHZ     25.2 DB     175 GR
10.00 KHZ     26.7 DB     174 GR
```




```
10 REM T-RESON
20dC1,1,2,50E-9
30 C2,2,4,5E-9
40 C3,4,5,5E-9
50 C4,3,0,10E-9
60 R1,2,3,100E3
70 R2,3,5,100E3
80 R3,4,0,47E3
90 R4,6,0,6.8E3
100 T1,5,2,0,40E-3,100,1E9
110 END
```

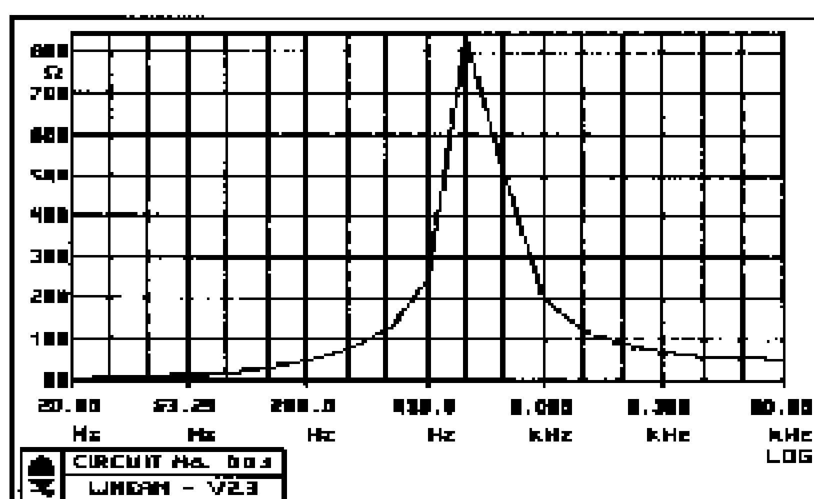


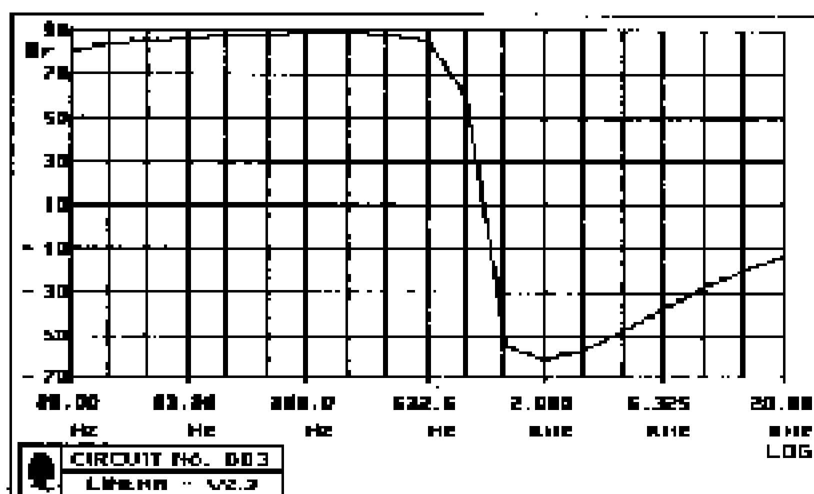


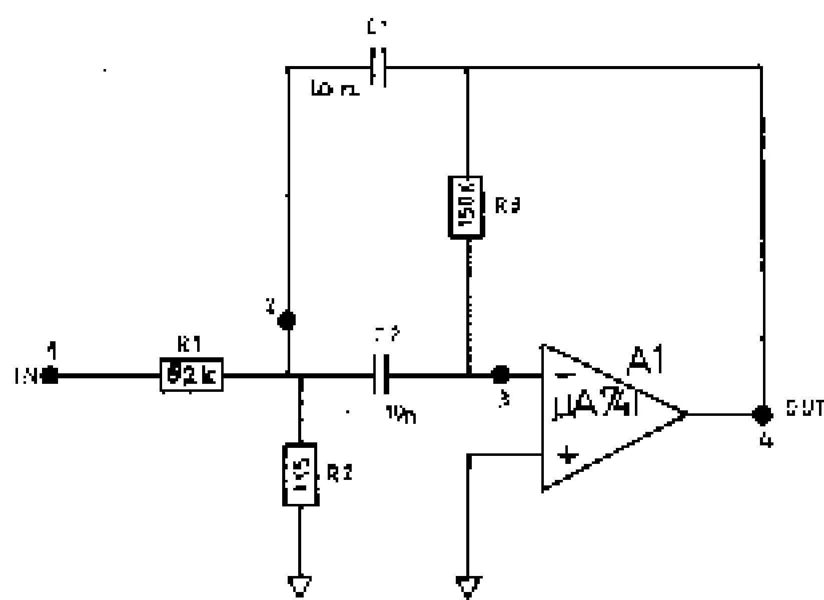
```
CIRCUITNAAM: BANDDOORLAAT
CIRCUIT NO.: 3

20.00  HZ   5.060  OHM   81 GR
29.38  HZ   7.387  OHM   84 GR
43.09  HZ   10.82  OHM   86 GR
63.25  HZ   16.88  OHM   87 GR
92.83  HZ   23.41  OHM   88 GR
136.3  HZ   34.67  OHM   89 GR
200.0  HZ   51.01  OHM   89 GR
293.5  HZ   79.62  OHM   89 GR
430.8  HZ   179.3  OHM   89 GR
632.5  HZ   246.3  OHM   86 GR
928.3  HZ   836.3  OHM   81 GR
1.363 KHZ   465.4  OHM  -55 GR
2.000 KHZ   202.6  OHM  -61 GR
2.936 KHZ   123.5  OHM   56 GR
4.308 KHZ   88.29  OHM   46 GR
6.326 KHZ   70.48  OHM  -37 GR
9.283 KHZ   61.47  OHM  -21 GR
13.63 KHZ   56.43  OHM  -20 GR
20.00 KHZ   54.77  OHM  -15 GR
```

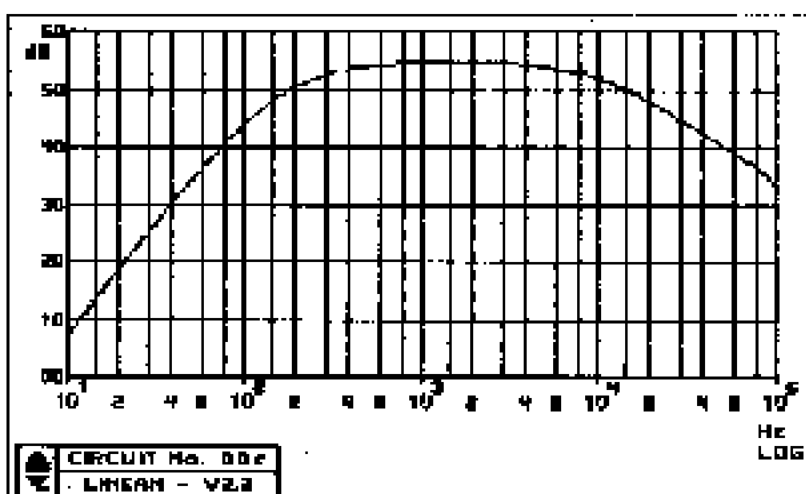
```
10 REM BANDDOORLAAT
20 R1,1,2,82E3
30 R2,2,0,1.5E3
40 R3,3,4,150E3
50 C1,2,4,10E-9
60 C2,2,3,10E-9
70 A1,0,3,4,2E5,3,75
80 END
```

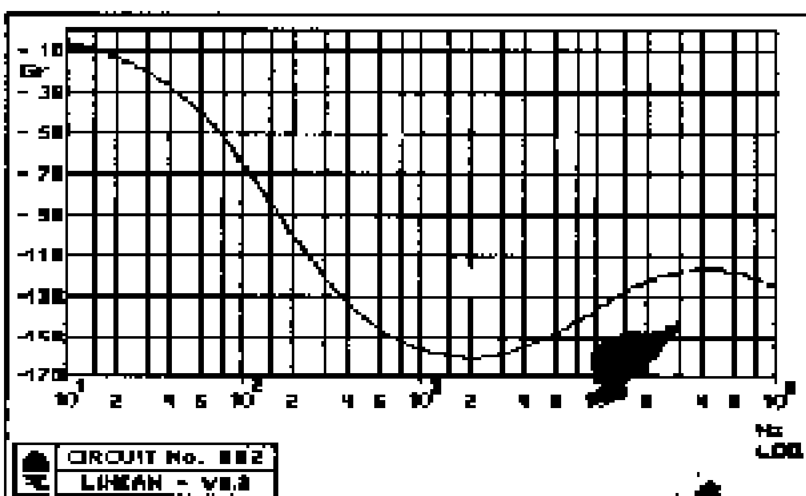


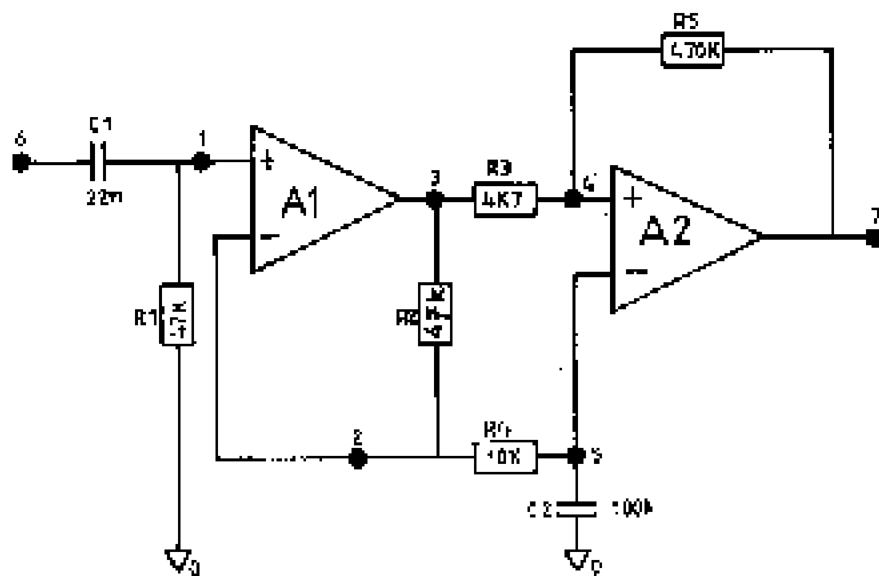




```
10 REM ATOMVERSTERKER
20 R1,0,1,47E3
30 R2,3,7,47E3
40 R3,3,4,47E2
50 R4,2,5,1E4
60 R5,4,7,47E4
70 C1,6,1,22E-9
80 C2,0,5,100E-9
90 A1,1,2,3,1E5,10,5
100 A2,4,6,7,1E6,10,5
110 END
```







DISYC

door Thijs Rutten

Het programma "disassembler" is overgenomen uit "ACORN USER" december 1983. Na "RUN" heeft het programma even tijd nodig om warm te draaien. Geduld dus !!!
Er zijn drie toetsen geprogrammeerd:

"SHIFT-Toets" zet het programma stop
"REPT -TOETS" zet het programma weer in actie
"SPACE-BAR" matigt de snelheid van het programma.

Op de vraag "STARTADRES" geeft men het begin adres op. (b.v. #2900). Het eindadres idem dito.
Geeft men als eindaddres 0 op, dan stopt het programma bij de eerstvolgende "BRK" of "JMP" of "RTS".
Als bijzonderheid zij vermeld dat met de "CTRL-TOETS" het adres waar de disassembler zich bevindt, met 80 verlaagd wordt.
Op dit verlaagde adres gaat het programma dan weer verder.
Door herhaaldelijk op de CTRL-Toets te drukken kan men dus steeds verder teruggaan met disassembleren.
Een gedetailleerde beschrijving van het programma is verkrijgbaar bij het drukwerkarchief.

R E G I O M E D E D E L I N G E N (Regio Limburg)

Let op : de regioavond voor de regio Limburg in AUGUSTUS komt te vervallen ivm vakantie.

Geplande demontraties regio Limburg:

Regioavond september: Uitgebreide demontratie RTTY en MORSE door J Colen en H Bastings

Regioavond oktober: aansturing LED martix met grafische tekens door J Feron.

```
*******************************************
*    Graphic dump voor Oki Microline 84.  *
*******************************************
```

door Jelle Aarnoudse

Het programma GDUMP84 is geschreven voor de Microline 84 van OKI. Dit is een tamelijk professionele A3 printer, die een jaar of 5 geleden in de professionele sector in grote getale is aangeschaft. Het valt dan ook te verwachten, dat de u84 binnenkort in ruime mate voor de hobbymarkt beschikbaar komt.
GDUMP84 is nog niet echt uit-ontwikkeld. Het werkt alleen voor CLEAR4, het heeft nog niet de vorm van een commando of statement, en het zou ook nog kunnen worden uitgebreid met opties als inverteren, verschuiven, e.d.

Wat kan GDUMP84 dan wel?
In essentie eigenlijk maar 1 ding: het op een matrix printer afdrukken van een CLEAR4-scherm. De enige optie wordt gevormd door het desgewenst in X- en/of Y-richting vergroten van de afdruk tot tweemaal de oorspronkelijke grootte.Het aanroepen gaat als volgt:

```
     X=2;Y=2;LINK #7000
```

Hierdoor wordt het scherm zowel in X- als in Y-richting tweemaal vergroot afgedrukt. Bij elke andere waarde van X en Y wordt het scherm 1:1 afgedrukt.

Er zit nog een eigenaardigheidje in de (mijn?) u84.
Wanneer men grafische karakters naar de u84 wil sturen, moet dat vooraf worden gegaan door de functiecodes: ESC.%.2.n1.n2.
Hierin vormt n1.n2 het aantal grafische karakters dat gaat worden gestuurd, waarbij n1 het aantal malen 128 en n2 het aantal malen 1 karakter weergeeft. Dus: als n1.n2=2.0, dan verwacht de u84 2*128 + 0*1=256 grafische karakters. Nou, dat komt dan mooi uit, want ons CLEAR4-scherm is 256 pixels breed! Mis. Voor alle andere combinaties van n1.n2 gaat dat op, maar niet voor 2.0! Zal wel een foutje in de software van de u84 zijn, dat mij de nodige avonden, hoofdbrekens, zware shaggies en wat dies meer zij, heeft gekost.
Hoe dit nu opgelost? Vrij simpel: we zeggen tegen de printer dat er 2.1 (dus 257) karakters komen, en sturen aan het eind van iedere regel een dummy-karakter mee. Kijk maar eens in de subroutines SEND'NEWLINE en SEND'EOL.

Dan is er nog wat. Bij een matrix-printer worden de onderste 1 of 2 printpennen nauwelijk gebruikt. Krijg je dus een afdruk waar in de zwarte vlakken na iedere 7 pixelrijen een horizontale witte lijn voorkomt, dan lijdt ook jouw printer aan dit euvel. Remedie: in de subroutine IPRINT na ESC.%.9 niet 16 maar 14 opgeven. Niet echt een oplossing, maar je ziet er geen "barst" van.

Tenslotte:

- ben ik in het bezit van een kopie van de User's Manual van de u84. Ik zal zorgen dat deze in het drukwerk-archief (Bruno Tossaint, red.) terecht komt, zodat deze ook voor anderen beschikbaar is.
- staan op de regioschijf naast GDUMP84 ook nog DEMO84 en OPEL84. Deze laatste heb ik gepikt van een eerdere regioschijf. DEMO84 geeft een demonstratie van de werking van GDUMP84.
- Om GDUMP84 te kunnen assembleren is SALFAA-2.0 nodig.
- Bij het ontwikkelen van dit programma heb ik dankbaar gebruik gemaakt van het artikel "GDUMP STAR SG-10" van Joop Engels, gepubliceerd in A.N. 5/3.

- Veel plezier!

Jelle Aarnoudse
Bekkerweg 72
6417 BW Heerlen
Tel: 045-710139

BASICODE RECHTSTREEKS NAAR DISK.

F.v.Helvoort.

Met behulp van de programma's "RADIOLADEN" en "BEWERKEN" zijn we in staat om basicode-programma's rechtstreeks van de radio op de disk te saven.
Eerst gaan we met het programma "RADIOLADEN" de files binnen laden van de radio, dit doen we rechtstreeks ,dus de uitgang van de radio gaat in plug SK2 van de computer (cas in).
Het programma runnen we op #1000 (eventueel aanpassen).
Bij de vraag "nummer van de file " vullen we 1 in.
Bij de vraag "laden vanaf de radio" drukken we op toets 1;
en ,als de begin-toon van het basicode-programma doorkomt, drukken we op de spatiebalk.
Als het programma binnen is ,wordt het meteen op disk opgeslagen en gaan we weer terug naar de vraag "radioladen of stoppen".
Zo kunnen we de programma's na elkaar binnen laden.
Ze worden opeenvolgend genummerd.
Als alles binnen is ,kunnen met behulp van het programma "BEWERKEN" de basicode-programma's omzetten naar BASIC.
De standaard-regels (tot aan regel 1000) staan van #2900 tot #2FFF ; regel 1000, het begin van het basicode-programma, begint op #3000.

Veel plezier met basicode,

F.v.Helvoort,
tel. 073-213691.

```
**********************************
* FORMATTED PROGRAM LISTING *
**********************************
```

door Roland Leurs

Dit is een machinecode utility is geschreven om lange listings van programma's gemakkelijk leesbaar te maken. Als het gerund wordt, voegt het programma het statement FLIST aan de P-Charme interpreter toe.

Lange en ingewikkelde regels met meerdere statements zijn meestal onduidelijk leesbaar in 32 of 40 kolommen mode. Met het statement FLIST worden deze multi-statement regels gesplitst en afgedrukt met ieder statement op een nieuwe regel.

Het statement verandert de WRCH-vector naar een eigen routine. Deze nieuwe schrijfroutine kijkt bij elk karakter of het een ; is. Zo ja, dan wordt er een linefeed geprint en vijf spaties. De ; wordt niet afgedrukt. Pas dus op als er een ; in een string staat. Deze wordt ook op deze manier verwerkt.

Verder is er ook nog een printer optie aanwezig. Als U de listing wil afdrukken op papier druk dan op de letter Y. Mocht de printer om de een of andere reden off line staan, dan gaat het programma gewoon zijn weg. Er blijft dus niets hangen.

De syntax van FLIST is hetzelfde als van het statement LIST. Tijdens het listen kunt U stoppen met escape. De printer wordt dan uitgezet en de WRCH-vector wordt weer hersteld.

Er worden ook nog drie zero page adressen gebruikt. U kunt deze zelf aanpassen in het programma. (Regel 240, variabele Z).

Tot slot nog een voorbeeld :

programmaregel:

```
   111P=#2800;[;LDA@7;JSR#FFF4;RTS;]
   222LINK #2800;END
```

met FLIST afgedrukt als:

```
   111P=#2800
        [
        LDA@7
        JSR#FFF4
        RTS
        ]
   222LINK #2800
        END
```

```
10 REM *********************************************
20 REM *                                           *
30 REM *           FORMAT PROGRAM LISTING          *
40 REM *                                           *
50 REM *        Geschreven door Roland Leurs       *
60 REM *                                           *
70 REM *     Dit programma is de source  van het   *
80 REM *     statement FLIST.  Dit statement zet   *
90 REM *     alle instructies van een prog.regel   *
100 REM *     onder elkaar.                         *
110 REM *                                           *
120 REM *     De syntax van het statement is het-   *
130 REM *     zelfde als die van LIST.              *
140 REM *                                           *
150 REM *                                           *
160 REM *       (c) 1988   S M U R F S O F T        *
170 REM *                                           *
180 REM *********************************************
190
200 PRINT $12''"*****************************"'
210 PRINT       "*  FORMAT  PROGRAM  LISTING  *"'
220 PRINT       "*****************************"'''
230 INPUT "Waar assembleren ",A
240 @=0;Z=#80;REM ZEROPAGE ADRESSEN: Z,Z+1,Z+2
250 DIM RR(5);FOR X=0 TO 5;RR(X)=#FFFF;NEXT X
260 FOR X=1 TO 2
270 PRINT "PASS ",X,$21
280 P=A
290 !P=#80C6E3FF;P=P+3
300 $P="FLIST";P=P+LEN P
310 ?P=A/256|#80;P?1=A%256+11
320 P?2=#80;P=P+3
330[ \ assembler format program listing
340:RR0   \start statement
350       JSR #F7D1
360];$P="PRINTER LISTING Y/N ?";P=P+LEN P
370[      NOP
380       JSR #FFE3
390       CMP @CH"Y"
400       BNE RR6
410       LDA #B001
420       BMI RR6
430       LDA @2;JSR #FFF4
440:RR6   LDA #209;STA Z+1
450       LDA #208;STA Z
460       LDA @(RR1%256);STA #208
470       LDA @(RR1/256);STA #209
480       JSR #FFED
490       JMP RR6
500:RR1   \schrijfvector aangepast
510       PHA
520       LDA #B001
530       CMP @#DF     \ESCAPE?
```

```
 540        BEQ RR4
 550        PLA
 560        CMP @CH";"
 570        BEQ RR2
 580        JMP (Z)
 590:RR2    JSR #FFED
 600        STX Z+3
 610        LDX @#04
 620        LDA @#20
 630:RR3    JSR #FFF4
 640        DEX
 650        BPL RR3
 660        LDX Z+3
 670        JMP (Z)
 680:RR4    \escape ingedrukt, einde routine
 690        LDA Z  ;STA #208
 700        LDA Z+1;STA #209
 710        JSR #FFED
 720        LDA @#03;JSR #FFF4
 730        JMP #C2CF
 740:RR5 \list routine
 750];REM COPY LIST-STATEMENT
 760 FOR Y=0 TO 123
 770 P?Y=Y?#CA61
 780 NEXT Y
 790 P=P+100;[;JMP RR4;]
 800 P=P+8;[;JMP P-82;]
 810 PRINT $6$13
 820 NEXT X
 830 PRINT "STATEMENT : FLIST  (syntax als bij LIST)."''
 840  PRINT "          ...  met de vriendelijke groeten  van Roland
...";'
 850 ?#3FC=A/256;END
```

Ook nu weer ...

... met de vriendelijke groeten van Roland ...

```
***********************************
*    E R A S E   en   M E N U     *
***********************************
```

door Jeroen van der Veen

Hier twee voorbeelden van nuttige GDOS utilities in een gebruiksvriendelijke verpakking. Nodig is dus een GDOS kaart en een VDU80, eveneens is P-charme vereist voor een goede werking van het programma. De werking van de programma's is simpel, ze werken beide op dezelfde manier. Het programma laat als help-functie alle mogelijkheden op de statusregel zien.

Eerst een algemene beschrijving:

Beide programma's gebruiken de volgende toetsen:

<@> kursor omhoog
</> kursor omlaag
<[> kursor links
<]> kursor rechts
<R> restart, programma start opnieuw en vraagt drivenummer
<Q> Quit, programma wordt beeindigd.

:MENU, een kort basic programmaatje, dat de Directory toont op het scherm, waarna de gebruiker het programma kan inladen door het met <return> te selecteren.

:ERASE, een programma, dat op dezelfde wijze werkt als MENU, men selecteert met <return> de te verwijderen programma's. Nadat men op <E> gedrukt heeft worden de files werkelijk gewist.

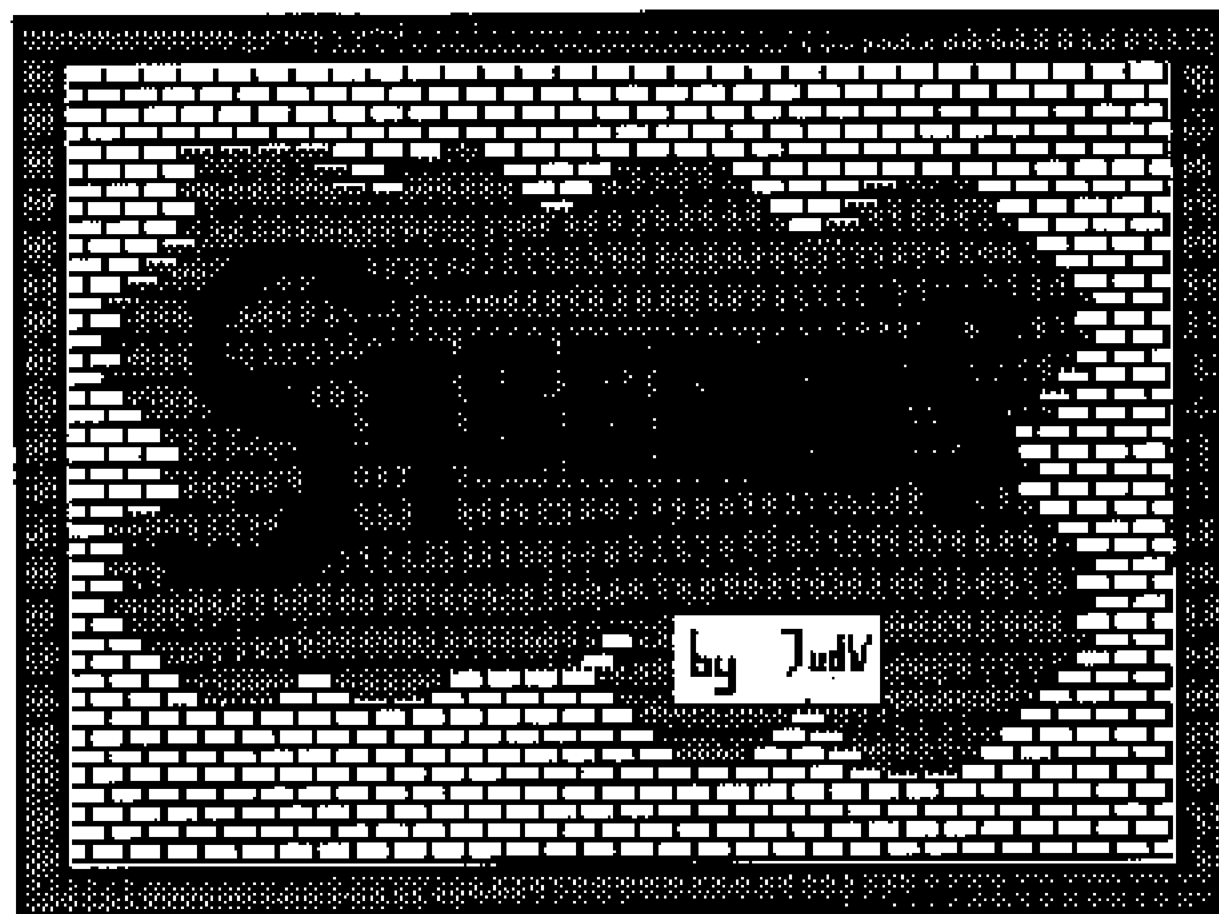
Nog veel plezier met deze hopelijk nuttige programmaatjes.

```
***********************************
*         S Q U E N I S           *
***********************************
```

door Jeroen van der Veen

Een (hopelijk) leuk spelletje voor de atom, geprogrammeerd in de vele vrije uren van de zomervakantie. Het spel is zeer simpel van opzet, maar wel zeer snel en (bij mij tenminste) zeer verslavend. Squennis doet sommige mensen misschien denken aan de eerste spelcomputer PONG, de grafische uitwerking is wel aardig wat beter (een plaatje zegt meer dan 3 woorden, dus zie het plaatje van het spel verderop). De bedoeling is de bal zo lang mogelijk tegen muur

en bat te laten ketsen.Elke keer als de bal het batje raakt krijgt men een punt. De hiscore wordt bijgehouden. Als de bal niet geraakt wordt verliest u een leven, en dat wordt u ook zeer duidelijk duidelijk gemaakt. Als uw drie levens op zijn, eindigt het spel en uw hiscore wordt in de jaarboeken bijgeschreven. Uw kunt door op de vuurknop te drukken weer verheropnieuwstarten.
Het programma maakt gebruikt van de GAGS-rom, en u heeft uiteraard een clubcompatible joystick (pretknuppel, lolstokje, etc.) nodig om dit programma naar behoren te laten verlopen. Bij het begin van het spel wordt er een picture ingeladen, dus het is wel prettig als u een disc-drive heeft. De naam is een samentrekking van, hoe raadt u het, squash en tennis.
Hierbij wens ik u nog veel speelplezier met SQUENIS.



```
*****************
* A . B . C . *
*****************
```

door Thijs Rutten

Het programma A.B.C. - ATOM BASIC CONDENSER - is gepubliceerd in YOUR COMPUTER van Februari 1983.
Na RUN verschijnt een menu. U kunt nu kiezen uit de mogelijkheden door het intoetsen van de geinverteerde karakters.
U kunt meerdere keuzes maken door meerdere karakters in een "commando string" te plaatsen. Zelfs eenzelfde letter meerdere malen lukt goed, hoewel dit weinig zinvol is.
Daarna wordt gevraagd naar de plaats van het in te korten programma, dat uiteraard in een andere textspace moet staan. Men geeft het most significant byte in, b.v. #29.
Dit is dus niet het A.B.C. programma zelf!!!!
Komt in de commando-string een ongeldig karakter voor, dan wordt dit aangegeven en gevraagd om een correctie. Men kan eventueel een geinverteerde 8 intoetsen, een blank dus , om een fout te corrigeren.
Bij het begin van elke routine gaat de bel. Drukt men nu op een toets dan wordt die routine overgeslagen.
De "WRAP ALL BASIC LINES" routine voegt basicregels samen tot een lengte van 254 karakters.
Dergelijk lijnen kunnen niet ge-edited maar wel geinterpreteerd worden !!
De laatste basic regel van het programma kan niet gelist worden.
Het laatste stukje van deze regel is een stukje machine code.
Het programma is niet geschikt om niet basic programma's onder handen te nemen. Het effect is wel voorspelbaar , maar niet datgene wat men beoogde !
De initialiserings-routine verandert alle regelnummers in het te bewerken programma in "0".
Daarom het programma tijdens het initialiseren niet onderbreken. Het zou een hopeloze puinhoop worden.
Wil men regelnummers niet wijzigen, omdat er b.v."goto's of "gosubs" in het programma voorkomen, die naar een bepaalde regel verwijzen, dan dient men deze regelnummers van een geinverteerd "@" te voorzien.
Bijvoorbeeld--------500@REM acorn atom etc------------
is een basicregel, die blijft behouden. Ook het "REM" statement zal niet worden weggewerkt!!!!
Is het programma in de terminale fase, dan geldt wat voor de initialisering is gezegd--- vingers van de toetsen, want de nieuwe regelnummers worden dan uitgedeeld!!!!

Good luck !!!

LISTER

door Leo Gijssel

Hierbij een klein en zeer simpel hulpprogramma voor down-loaden van listings vanuit ATOMTEL.
Het print strings uit die in een bepaald deel van het geheugen staan. Als de volgende string begint met #FF, of steeds na het printen van een in te stellen aantal regels, stopt het programma.
Wanneer ik naar Atomtel bel en een listing opvraag kan ik de binnenkomende data kopiëren met de logfunctie. Bijvoorbeeld met het programma RSCOM, zie AN.89-1.
Een programma komt dan in ASCII in het geheugen te staan inclusief wat rotzooi, bijvoorbeeld extra spaties en loze regels, zoals de melding " druk op een toets .....", om aan te geven dat er nog meer beelden volgen. Dat hebben we achter niet nodig, wat we willen is een BASIC-programma maken van deze ASCII-dump.
Wanneer het scherm vol is stopt het programma, de variabele A bevat het adres van de volgende te printen string. Door nu met de cursor alle programmaregels op het scherm te kopiëren, kan de listing worden ingelezen. Wanneer alle regels op het scherm gekopieerd zijn, kan LISTER weer worden opgestart met de instructie GOTO Z of G.Z en kan de volgende pagina gekopieerd worden. Omdat in A nog steeds het adres staat van de volgende string, krijg je automatisch meteen de volgende pagina.
De variabele C dient als teller van het aantal geprinte regels. Op deze wijze is op een zij het arbeidsintensieve manier een programma te verkrijgen, hoewel het toch nog flink wat sneller gaat dan overtypen. Een BASIC-programma van 1 a 2 K is in notime gekopieerd. Langere programma's zijn meestal meer in trek en men heeft er dus wat meer werk voor over, toch wens ik het niemand toe om op deze wijze de source van bijvoorbeeld BRANQUART te moeten bewerken.
De kans op fouten maken is vrij gering, het is een kwestie van de copy-toets ingedrukt houden, geen regels vergeten en er een beetje opletten geen spaties achter de programmaregels mee te nemen, het programma wordt daardoor onnodig lang en ook trager.
Met LISTER op £2900 is een programma van bijna 10K te laden, als de logfunctie op £5000 begint.
De oplettende lezer zal meteen begrijpen waarom LISTER zulke hoge regelnummers bezit. Wanneer U de hele listing van het programma kopieert, staat LISTER (regel 20000 en verder) er netjes achter. P-Charme voorzetten, DEL 20000, geven en klaar is Uw programma !
Zolang er geen echt conversieprogramma wordt gepubliceerd, is het hier gebodene m.i. een redelijke noodoplossing.
Bij dit programma heb ik een dump gevoegd die ik zelf uit ATOMTEL gehaald heb. Deze file heet dan ook DUMP en het is de bedoeling dat hiermee uitgetest kan worden hoe alles werkt.
Laad LISTER op £2900 en laad DUMP op £5000 (met *LOAD DUMP), geef een RUN en kijk wat er gebeurt ! Succes !

ED80.v4

een versie voor de 80-koloms Elektuur-kaart, door Dick Protzman

Tot mijn verrassing las ik in ATOM NIEUWS 7.2 blz.63/64. een artikel over de editor, en dat deze in assemblervorm ter beschikking was gekomen, door o.a. het goede werk van G.vd Brandhof, die er zijn eigen uitbreidingen bij gemaakt heeft, en wel : .h - .b - en .u commando's. Daar deze uitbreidingen speciaal bedoeld waren voor een BROTHER printer en een andere VDU driver had ik hier niet zoveel aan, daar ik een EPSON LX800 printer bezit en een ELEKTUUR 80 kolomskaart. Maar de eerste opzet was goed, en deze heb ik dus overgenomen.
Het .h commando heb ik gehouden, en daarbij de printer uitbreidingen standaard in het programma opgenomen. dat wil zeggen dat er geen aan en uit commando meer is voor de speciale printer mogelijkheden. Ook heb ik de speciale tekens die de printer in onderline, italic, double-width of emphasized zetten veranderd in meer voor zich sprekende karakters. Dit alles heeft wel tot gevolg dat men deze tekens niet meer (zonder meer) in een tekst kan plaatsen, maar omdat deze tekens toch niet zoveel worden gebruikt, dacht ik niet. dat het, in deze, moeilijkheden veroorzaakte.

TOT SLOT :
Deze source is uitsluitend bedoeld voor ELEKTUUR 80 kolomskaarten, en voor die mensen die dus deze source willen gebruiken..., vergeet niet de waarden aan te passen die voor Uw computer systeem gelden, en deze staan in de remark regels door het hele programma vermeld. Ik wens de gebruikers een plezierige tik morgen, middag of avond (geen nacht) toe.

De commando's die erbij gemaakt zijn kunt U terugvinden op de volgende twee pagina's. Bij deze nog dank aan G.J. Noorland.

Groetjes van Dick Protzman dipro'88

EDITOR COMMANDO'S

EDRO.v4

- A	(After)	Voegt tekst in na cursor, en druk op {COPY} toets.
- B	(Before)	Voegt tekst in voor cursor, en druk op {COPY} toets.
- C	(Copy)	Copieert tekst tussen {@} en indruk {C} toets, in de buffer.
- E	(Enter)	Tekst wordt via {COPY} toets in geheugen gezet.
F	(Find)	Zoekt tekst vanaf huidige scherm, bvb, F/tekst/ of F/tekst/vervang tekst/a Antwoorden met {Y/N} - stoppen met {ESC} toets. {a} = vervangen van tekst zonder {Y/N} mogelijkheid.
- G	(Getret)	Aan/uit van returns zichtbaar maken op het scherm.
H	(Home)	Zet cursor links bovenin scherm.
- I	(Insert)	Voegt karakter in ingetypt na {I} voor de cursor.
- L	(List)	List de hoeveelheid aan pagina's die nodig zijn om, de tekst uit te printen, afhankelijk van {.l}, en {.r}. Standaard {50} regels lang en {70} karakters breed. Dit commando werkt niet goed samen met uitvullen.
- N	(Next)	Naar het volgende scherm met tekst.
- O	(Option)	Printer/scherm keuze. {P}=printer - {Toets}=scherm, afbreken met {ESC} toets. Stop door {SHIFT} toets - verder door {CTRL} toets,
- P	(Previous)	Naar vorige scherm met tekst.
Q	(Quit)	Verlaat de tekst editor.
- R	(Replace)	Vervangt de tekst tussen {@} en {R} toets, direct daarna nieuwe tekst invoeren.
- S	(Start)	Ga naar begin van de tekst file.
- T		Copieert tekst tussen {@} en {T} toets, in de buffer en delete daarna deze tekst.
W	(Where)	Geeft het adres van de tekst-end marker.
- X		Verandert het teken onder de cursor in het direct, na de {X} ingetoetste teken. {X+COPY} wist alles na de cursor.
- Z		Ga direct naar het laatste deel van de tekst.
*		Maakt {DOS/COS} commando's mogelijk.
- >		Tekst naar cassette/floppy >filenaam.
- <		Tekst van cassette/floppy <filenaam.
- @		Tekst marker - opheffen met {ESC} toets.
- ?	(Help)	Geeft een help regel voor printer commando's, en uitbreidingen op de standaard {EDIT} box.
- ^		Scrolt de tekst {1} regel naar boven.
- (Haken)		De grote {HAKEN} verplaatsen de cursor direct, naar begin of einde van een regel tekst.
(Delete)		De {DELETE} toets delete {1} karakter onder de, cursor, of delete het stuk tekst tussen de marker, {@} en de {CURSOR}.
- (*R.txtname)		Ook kunt U met bvb {*R.txtname} direct opstarten, mits U de txt.file met deze box gesaved heeft, en de juiste box voorstaat.

WIST U WEL !

Druk na elke tekst invoer op de {COPY} toets.

EDITOR PRINTER CODES

EDBO.v4

- .a(nr.) Nieuwe pagina of met {.an} kan men (n) regels, op dezelfde, pagina definiëren.
- .c Voor centreren tekst in (1) regel, bvb bij (d.width?, {.r70}-{20} karakters breed = {.r50} tijdelijk.
- .d Dubbele returns worden gegenereerd, standaard is {1} return.
- .e Paginanummer in het midden onderaan afdrukken.
- .h(kar) Karakter achter {.h} wordt een spatie, doch wordt bij uitlijnen niet als zodanig gezien, standaard het {!} karakter.
- .i(nr.) Inspringen van {n} spaties, standaard geen inspringen.
- .j Zorgt voor een rechte kantlijn door uitvullen van de tekst, standaard is uitvullen.
- .k Drukt af per vel papier, men krijgt dan (PAPER), en er word gewacht op een toets indruk, afhankelijk van {.l} commando.
- .l(nr.) Aantal regels die U op een vel papier wilt hebben, standaard {58} - 11 inch, 12 inch = {61} regels.
- .m Inspringen is {0} voor {1} regel.
- .n Schakelt {.j} commando weer uit, dus niet uitvullen.
- .o(nr.) Stuurt controle karakter naar printer. bvb, {.o27.o15} = select condensed printing.
- .p(nr.) Start met paginanummering op, bvb {nr.1}, maximaal = {255}. {.p0} zet paginanummering uit, standaard geen paginanummering.
- .r(nr.) Regel lengte veranderen, standaard {70} karakters.
- .s Enkele returns worden gegenereerd, standaard is {1} return.
- .t(nr.) Inspringen van {1} regel met (n) spaties.
- .w(nr.) Waar paginanummering {rechterkant}, meer naar het midden of, naar buiten, standaard waarde = (70), karakters, dit is *onafhankelijk* van het {.r} commando.
- .x(kar) Verandert controle teken in {kar}, standaard {.} karakter.
- .\ Commentaar - wordt overgeslagen, dus *niet* afgedrukt.

WIST U WEL :

Dat de printer controle tekens alleen werken, als men ze vooraan een regel plaatst, na een return, of direct achter elkaar zet, bvb, {.l66.r80.h!}.

- .l Dat U altijd met {.l} moet starten daar hier alles op, de standaard waarden ingesteld word, Daarna kunt U de andere {.} codes gebruiken.
- .p1 Dat de pagina nummering op het scherm per {256} bytes is

Printer bewerkings commando's, die bij elkaar horen zijn : {.l + .r} {.a + .k} {.s + .d} {.j + .n} {.p + .o + .w} {.i + .m + .t}

Een nieuwe printroutine,het 8e-printerbit toch nog anders

H.Bastings.

Ikzelf ben in het bezit van een racksysteem,dat voornamelijk bestaat uit wire-wrap printen.
Van veel belang is dit niet ,alleen ,het is de oorzaak van dit verhaal.
Het gebeurde aldus in het jaar 1989,(op de heks moet je nog even wachten,),dat ik besloot ,dat het systeem waarop ik 99% van mijn ATOM-proefjes doe, toch ook wel voorzien mocht zijn van de mogelijkheid deze gedachten-kronkels op papier te zetten.

Zoals velen voor mij al was opgevallen,moest ik mij tevreden stellen met 7 bits, maar wilde toch ook het 8e bit gebruiken.
Eerst dan maar de software napluizen.
Hier vinden we het volgende :

```
FEFB   PHA            ZET TE PRINTEN KARAKTER OP STACK
FEFC   CMP @#02       IS HET #2
FEFE   BEQ #FF27      JA,DAN GA PRINTER INSCHAKELEN
FF00   CMP @#03       NEE,IS HET DAN $3
FF02   BEQ #FF38      JA,DAN ZET DE PRINTER UIT
FF04   CMP #FE        KARAKTER IN #FE,DAN NIET PRINTEN
FF06   BEQ #FF36      EN NAAR EINDE ROUTINE
FF08   LDA #B80C      LAAD PCR
FF0B   AND @#0E       BEKIJK BIT 1 T/M 3
FF0D   BEQ #FF36      0,DAN STAAT PRINTER UIT,EINDE
FF0F   PLA            HAAL KAR. VAN DE STACK
FF10   BIT #B801      BIT PA7 HOOG? (=BUSY)
FF13   BMI #FF10      DAN WACHTEN TOT NIET MEER BUSY
FF15   STA #B801      ZET NU MAAR IN OUTPUT-POORT (PA0-PA6)
FF18   PHA            KARAKTER WEER OP STACK
FF19   LDA #B80C      LAAD PCR
FF1C   AND @#F0       LAAT BIT'S VOOR B-POORT ONGEMOEID
FF1E   ORA @#0C       SET BIT 2 EN 3 (CA2 LAAG-STROBE )
FF20   STA #B80C      ZET IN PCR
FF23   ORA @#02       SET BIT 1
FF25   BNE #FF33      NAAR CA2 HOOG ZETTEN
\ ROUTINE SCHAKELT PRINTER AAN
FF27   LDA @#7F       SET BIT 0 T/M 6 VAN A-POORT
FF29   STA #B803      OP OUTPUT ( ZET IN DDRA )
FF2C   LDA #B80C      LAAD PCR
FF2F   AND @#F0       LAAT BIT 4 T/M 7 MET RUST
FF31   ORA @#0E       SET BIT 1 T/M 3
FF31   STA #B80C      ZET IN PCR (CA2 HOOG EN PRINTER AAN )
FF36   PLA            HAAL KARAKTER TERUG VAN STACK
FF37   RTS            EINDE ROUTINE
\ROUTINE ZET PRINTER UIT
FF38   LDA #B80C      LAAD PCR
FF3B   AND @#F0       BIT 0 T/M 3 OP NUL
FF3D   BCS #FF33      ZET IN PCR (PRINTER UIT )
```

WIE vindt hier het gebruik van de acknowledge-lijn in terug ??

Ik niet en jullie waarschijnlijk ook niet ,naar mij lijkt.!
Deze lijn wordt dus helemaal niet gebruikt door het systeem.
De software kijkt standaard alleen maar naar de busylijn,die PA7 in beslag neemt.
Mocht het ons nu lukken alleen de acknowledge-lijn te bemonsteren,dan komt PA7 ter beschikking voor de verwerking van het 8e-printerbit.!
De navolgende routine verzorgt dit geheel; past op de plaats van de oude routine en laat zelfs nog 2 byte's over.

start #FEFB   printer aan #FF27   printer uit #FF38

En dan nu de nieuwe routine :

```
FEFB    PHA             BEWAAR KARAKTER OP STACK
FEFC    CMP @#02        IS HET $2  (PRINTER AAN )
FEFE    BEQ #FF27       JA,DAN GA PRINTER INSCHAKELEN
FF00    CMP @#03        NEE,IS HET DAN #3
FF02    BEQ #FF38       JA,DAN ZET DE PRINTER UIT
FF04    CMP #FE         VERGELIJK MET ?FE,DAN KARAKTER NIET PRINTEN
FF06    BEQ #FF1B       GELIJK, DAN NAAR #FF30 (EINDE ROUTINE )
FF08    LDA #B80C       LAAD VIA PCR
FF0B    AND @10         PRINTER AAN ?
FF0D    BEQ #FF1B       BIT 2 EN 3 NUL, DAN STAAT PRINTER UIT
FF0F    PLA             HAAL KARAKTER TERUG
FF10    PHA             BEWAAR
FF11    STA #B801       ZET IN DATA-REGISTER A
FF14    LDA #B80D       LAAD VIA IFR,ACKN GEWEEST ?
FF17    AND @#02        OP CA1
FF19    BEQ #FF14       ZONIET ,DAN WACHT
FF1B    LDA #B80F       RESET CA1-VLAG
FF1E    PLA             HAAL KAR.TERUG
FF1F    RTS             EINDE ROUTINE
FF20    STA #B80C       ZET PCR
FF23    BCS #FF1B       SPRING ALTIJD
FF25    NOP
FF26    NOP
FF27    LDA @#FF        LAAD FF
FF29    STA #B801       IN DDRA (PA0-PA7 OUTPUT )
FF2C    LDA #B80C       LAAD PCR
FF2F    AND @#F0        GEEN VERANDERING IN B-POORT
FF31    ORA @#0A        ZET CA2 IN PULSE MODE
FF33    STA #B80C       ZET PCR
FF36    BNE #FF1B       NAAR EINDE ROUTINE
FF38    LDA #B80C       LAAD PCR
FF3B    AND @#F0        GEEN VERANDERING IN B-POORT
FF3D    BCS #FF20       ZET PRINTER UIT
```

Nu de aansluiting naar pen 11 van de centronics connector omleggen naar pen 8,en het is klaar.
Voor degenen die niet het risico willen lopen de 6522 te beschadigen ,is het aanbevelenswaardig de PA7-lijn eerst te bufferen met behulp van 2 secties van een LS04 in serie,of een sectie van een 7402 die op de rug van de nabijgelegen LS00 geplaatst kan worden.

Veel succes,
H.Bastings.
Regio Limburg

Wilt u lid worden van de ATOM COMPUTER CLUB?
Neem dan contact op met de penningmeester van de regio waar u bij ingedeeld wilt worden. Deze kan u inlichten omtrent het lidmaatschap.

Regio OVERIJSSEL/GELDERLAND;
Zie regio Arnhem

Regio TWENTE;
G.J.Noorland Prinses Ireneweg 4 7433 DE Schalkhaar
05700-25294

Regio NOORD-HOLLAND;
P. van Kuik Zuideinde 54-a 1843 JP Groot-Schermer
02997-1902

Regio DEN HAAG;
Th.Waayer L.Couperusstraat 6 2274 XP Voorburg
070-862504

Regio DELFT;
Zie regio Den Haag

Regio ROTTERDAM;
Zie regio Den Haag

Regio CENTRUM;
P.van Mourik Ruiterstede 60 3431 XN Nieuwegein
03402-48781

Regio ARNHEM;
J.Hartog Keyenbergseweg 60 6871 WK Renkum
08373-13757

Regio ZEELAND;
E.Gijssel Ruysdaelstraat 6 4462 AD Goes
01100-32557

Regio BRABANT-OOST;
P.Ehrlig Roostenlaan 266 5644 BS Eindhoven
040-114183

Regio LIMBURG;
J.Colen Provincialeweg Z-27 6438 BA Oirsbeek
04492-1957

Regio BELGIE;
Zie Regio Limburg

Bij het aangaan van het lidmaatschap kunt u de contributie overmaken op de rekening van de federatie. Vermeld hierbij uw volledige naam, adres en de regio waar u bij ingedeeld wilt worden.